ESCUELA POLITECNICA NACIONAL
FACULTAD DE INGENIERIA ELECTRICA

PROGRAMA PARA VISUALIZAR LA COMUNICACION DE UN MICROCONTROLADOR CON SUS PERIFERICOS

TESIS PREVIA A LA OBTENCION DEL TITULO DE INGENIERO EN LA ESPECIALIZACION DE ELECTRONICA Y TELECOMUNICACIONES

CODIGO FUENTE

FRANZ MISAEL ESPARZA ROMERO
PATRICIO XAVIER QUINTANILLA PEÑA

MARZO DE 1999

ANEXO A

CÓDIGO FUENTE

Form2

```
'Declaración de variables
Dim sMatriz_Opcode(43) As String 'Contiene el Opcode  de todas las
                'instrucciones que trabajan con el
                'microcontrolador 8031
Dim sMatriz_Operando12(20) As String   'Contiene los posibles operandos
                        'que forman parte de una instrucción
Dim FileNumber1 As Integer 'Contendrá el siguiente número de archi vo que le entrege Freefile
Dim Y As Integer, Z1 As Integer, Z2 As Integer, Z3 As Integer, Z4 As Integer
'Variables para manejar la instruccion MOV
Dim sVariantesInsMOV(40) As String  'Contiene los caracteres de los operandos de
                        'la instruccion MOV
Dim sOperandosMOV   As String   'Guardara los caracteres leidos con LetrasLeidas
                'para compararse con los cargardos en la matriz
                'sVariantesInsMOV
Dim mv  As Byte 'Contador para realizar seleccion de bancos
'Variables para manejar la instruccion XCH
Dim ByteXCH As Byte 'Esta variable la utilizamos para guardar
                'momentaneamente el byte de uno de los operandos
                'de esta instruccion en formato decimal
Dim iNum_Data1 As Byte
Dim iNum_DataB(7) As Byte  'Matriz de 8 bits para guardar en forma binaria
                'el #Data leido
Dim UbiBanco As Byte    'Variable utilizada pa saber en que banco estamos
                'cuando se hagan instrucciones que contengan como
                'operandos #Data con registros Rn y @Ri
'Variable utilizadas en ejecucion de instrucciones ADD, subrutina EjecucionADD
Dim BitADD As Byte
Dim BitSobra As Byte
Dim Add As Byte 'Se usa para hacer ciclos for
'Variable utilizadas en ejecucion de instrucciones SUBB, subrutina EjecucionSUBB
Dim BitSUBB As Integer
Dim BitBorrow As Integer
Dim Subb As Byte 'Se usa para hacer ciclos for
'Variables Matriciales para almacenar las etiquetas y manejar los nombres
'especiales de los SFRs, registros de función especial
Dim sEQU_diDa(29, 1) As String  'Contiene la parte alfanumérica de la etiqueta,
                'los caracteres (columna 0) y la parte numérica
                'de la etiqueta (columna1) . Con un máximo de 30 filas
Dim sSFR_Operandos(19, 1) As String   'Contiene operandos, los registros de función especial
                        'como bytes y con sus nombre especiales (columna 0)
                        'Contiene la dirección de los SFR como datos decimales
                        '(columna 1)
Dim sSaltos_Rel(100)  As String  'Contiene las etiquetas rel para las instrucciones
                'de salto, y llamadas con retornos de subrutinas e
                'interrupciones
Dim iSaltos_Rel(100) As Integer  'Contiene la "dirección de ROM" de las instrucciones
                'a ser ejecutadas dependiendo de la etiqueta rel
```

```
Dim sSFR_Bits(55, 1) As String  'Para uso de instrucciones a nivel de bits
                                'en los SFRs

'Variables utilizadas en la subrutina Ubicacion_Bit
Dim iOperando As Byte   'Toma los valores de 2 o 3, que permiten
                        'saber si es el primer operando o el segundo
                        'respectivamente
Dim iNum_Byte As Byte   'Guarda el valor (decimal) de la ubicación del byte en la
                        'memoria Ram interna
Dim iNum_Bit As Byte    'Guarda el valor (decimal) de la ubicación del bit en la
                        'memoria Ram interna
'Variable para guardar la dirección de la última instrucción
'ejecutada antes de ir a antender la interrupción
Dim iSaltoInterrupcion_Timer0 As Byte
Dim iInterrupcion_Timer0 As Boolean  'Para saber que se esta ejecutando
                                     'la subrrutina de atención a la
                                     'interrupción
Dim iSaltoInterrupcion_Timer1 As Byte
Dim iInterrupcion_Timer1 As Boolean
Dim iSaltoInterrupcion_EXTI0 As Byte
Dim iInterrupcion_EXTI0 As Boolean
Dim iSaltoInterrupcion_EXTI1 As Byte
Dim iInterrupcion_EXTI1 As Boolean
Dim iAumento_Timers As Boolean  'Para relaizar el aumento de los Timers
                                'de ciclo en ciclo o todos los ciclos de máquina
                                'de cada instruccion, dependiendo si el usuario
                                'elige el modo Con Animación o Sin Animación
                                'respectivamente
'Variables para el controlde prioridad de las interrupciones
Dim INT0P As Byte   'Prioridad interrupción externa 0
Dim Timer0P As Byte   'Prioridad interrupción timer 0
Dim INT1P As Byte   'Prioridad interrupción externa 1
Dim Timer1P As Byte   'Prioridad interrupción timer 1
Dim saPrioridad(1, 3) As Byte    'Para controlar prioridad de
                                 'interrupciones
Dim EA As Byte  'Variable de comparación
Dim sSaltosVerdaderos(13) As String

Sub Atencion_Interrupciones()
Dim Pri1 As Byte
Dim EA1 As Byte 'Variable de comparación
Prioridad_Interrupcion
For Pri1 = 0 To 1
   For EA = 0 To 3
      saPrioridad(Pri1, EA) = 5   'Pues mayor prioridad tienen
                                  'números inferiores
   Next EA
Next Pri1
EA = 0
If iMemoriaRAMint(168, 0) = 1 Then  'Si el bit EA esta habilitado
   Pri1 = 0
   If iMemoriaRAMint(168, 7) = 1 And iMemoriaRAMint(136, 6) = 1 Then
      'Si esta el bit EX0 en 1 para que pueda
      'ser atendida la interrupción externa 0
      'Si se activo la bandera de la interrupción
      'externa 0 IE0
      saPrioridad(0, Pri1) = INT0P
      saPrioridad(1, Pri1) = 1
      Pri1 = Pri1 + 1
      EA = 1
   End If
   If iMemoriaRAMint(168, 6) = 1 And iMemoriaRAMint(136, 2) = 1 Then
      'Si esta el bit ET0 en 1 para que pueda
      'ser atendida la interrupción del Timer 0
      'Si se activo la bandera de desbordamiento
      'del Timer 0 TF0
      saPrioridad(0, Pri1) = Timer0P
      saPrioridad(1, Pri1) = 2
      Pri1 = Pri1 + 1
      EA = 1
```

```
End If
If iMemoriaRAMint(168, 5) = 1 And iMemoriaRAMint(136, 4) = 1 Then
   'Si esta el bit EX1 en 1 para que pueda
   'ser atendida la interrupción externa 1
   'Si se activo la bandera de la interrupción
   'externa 1 IE1
   saPrioridad(0, Pri1) = INT1P
   saPrioridad(1, Pri1) = 3
   Pri1 = Pri1 + 1
   EA = 1
End If
If iMemoriaRAMint(168, 4) = 1 And iMemoriaRAMint(136, 0) = 1 Then
   'Si esta el bit ET1 en 1 para que pueda
   'ser atendida la interrupción del Timer 1
   'Si se activo la bandera de desbordamiento
   'del Timer 1 TF1
   saPrioridad(0, Pri1) = Timer1P
   saPrioridad(1, Pri1) = 4
   Pri1 = Pri1 + 1
   EA = 1
End If
If EA = 0 Then
   Exit Sub   'Salimos de la subrrutina si ninguna interrupción
              'se ha habilitado
End If
'Determinamos de las interrupciones habilitadas, cual de
'ellas tiene mayor prioridad
EA1 = saPrioridad(0, 0)
Prij = 0
For Pri = 1 To 3 'Maximo 3 comparaciones
   If EA1 < saPrioridad(0, Pri) Then
      EA = saPrioridad(1, Prij)
   Else
      EA1 = saPrioridad(0, Pri)
      Prij = Pri  'Son iguales pero solo para esta condición
                  'de comparación
      EA = saPrioridad(1, Prij)
   End If
Next Pri
Select Case EA
   Case 1
      If iMemoriaRAMint(168, 7) = 1 Then  'Si esta el bit EX0 en 1 para que pueda
                                          'ser atendida la interrupción externa 0
         If iMemoriaRAMint(136, 6) = 1 Then  'Si se activo la bandera de la interrupción
                                             'externa 0 IE0
         'Saltamos a las localidades de la subrrutina de atención a la interrupción
            If iInterrupcion_EXTI0 Then   'Si ya esta ejecutandose la subrutina
               Exit Sub                   'de atención a la interrupción.
            Else   'Caso contrario saltamos a las localidades de la subrrutina de atención a la interrupción
               iSaltoInterrupcion_EXTI0 = ejec1
               iInterrupcion_EXTI0 = True
                     ejec1 = 2 - 1   "
                     iFilasSP = iFilasSP + 1
                     iMatrizSP(iFilasSP) = iSaltoInterrupcion_EXTI0
                     i16Bits1 = Val(sInstruccionesOperandos(iSaltoInterrupcion_EXTI0 + 1, 25))
                     ConversionDPTRD_B
                     For ilcall1 = 0 To 7
                        iMemoriaRAMint(SP + 2, ilcall1) = iBitDPTR(ilcall1)
                        iMemoriaRAMint(SP + 1, ilcall1) = iBitDPTR(ilcall1 + 8)
                     Next ilcall1
                     ActualizacionRAM
                     SP = SP + 2
                     iMemoriaRAMintD(129) = SP
                     ActualizacionRAMD_BH
            End If
         End If
      End If
   Case 2
      If iMemoriaRAMint(168, 6) = 1 Then  'Si esta el bit ET0 en 1 para que pueda
                                          'ser atendida la interrupción del Timer 0
         If iMemoriaRAMint(136, 2) = 1 Then  'Si se activo la bandera de desbordamiento
```

```
                    'del Timer 0 TF0
        If iInterrupcion_Timer0 Then   'Si ya esta ejecutandose la subrutina
          Exit Sub                'de atención a la interrupción
        Else   'Caso contrario saltamos a las localidades de la subrrutina de atención a la interrupción
          iSaltoInterrupcion_Timer0 = ejec1
          iInterrupcion_Timer0 = True
              ejec1 = 9 - 1 "
              iFilasSP = iFilasSP + 1
              iMatrizSP(iFilasSP) = iSaltoInterrupcion_Timer0
              i16Bits1 = Val(sInstruccionesOperandos(iSaltoInterrupcion_Timer0 + 1, 25))
              ConversionDPTRD_B
              For ilcall1 = 0 To 7
                iMemoriaRAMint(SP + 2, ilcall1) = iBitDPTR(ilcall1)
                iMemoriaRAMint(SP + 1, ilcall1) = iBitDPTR(ilcall1 + 8)
              Next ilcall1
              ActualizacionRAM
              SP = SP + 2
              iMemoriaRAMintD(129) = SP
              ActualizacionRAMD_BH
        End If
      End If
    End If
  Case 3
    If iMemoriaRAMint(168, 5) = 1 Then  'Si esta el bit EX1 en 1 para que pueda
                        'ser atendida la interrupción externa 1
      If iMemoriaRAMint(136, 4) = 1 Then  'Si se activo la bandera de la interrupción
                          'externa 1 IE1
      'Saltamos a las localidades de la subrrutina de atención a la interrupción
        If iInterrupcion_EXTI1 Then   'Si ya esta ejecutandose la subrutina
          Exit Sub                'de atención a la interrupción
        Else   'Caso contrario saltamos a las localidades de la subrrutina de atención a la interrupción
          iSaltoInterrupcion_EXTI1 = ejec1
          iInterrupcion_EXTI1 = True
              ejec1 = 16 - 1 "
              iFilasSP = iFilasSP + 1
              iMatrizSP(iFilasSP) = iSaltoInterrupcion_EXTI1
              i16Bits1 = Val(sInstruccionesOperandos(iSaltoInterrupcion_EXTI1 + 1, 25))
              ConversionDPTRD_B
              For ilcall1 = 0 To 7
                iMemoriaRAMint(SP + 2, ilcall1) = iBitDPTR(ilcall1)
                iMemoriaRAMint(SP + 1, ilcall1) = iBitDPTR(ilcall1 + 8)
              Next ilcall1
              ActualizacionRAM
              SP = SP + 2
              iMemoriaRAMintD(129) = SP
              ActualizacionRAMD_BH
        End If
      End If
    End If
  Case 4
    If iMemoriaRAMint(168, 4) = 1 Then  'Si esta el bit ET1 en 1 para que pueda
                        'ser atendida la interrupción del Timer 1
      If iMemoriaRAMint(136, 0) = 1 Then  'Si se activo la bandera de desbordamiento
                          'del Timer 1 TF1
      'Saltamos a las localidades de la subrrutina de atención a la interrupción
        If iInterrupcion_Timer1 Then   'Si ya esta ejecutandose la subrutina
          Exit Sub                'de atención a la interrupción
        Else   'Caso contrario saltamos a las localidades de la subrrutina de atención a la interrupción
          iSaltoInterrupcion_Timer1 = ejec1
          iInterrupcion_Timer1 = True
              ejec1 = 23 - 1 "
              iFilasSP = iFilasSP + 1
              iMatrizSP(iFilasSP) = iSaltoInterrupcion_Timer1
              i16Bits1 = Val(sInstruccionesOperandos(iSaltoInterrupcion_Timer1 + 1, 25))
              ConversionDPTRD_B
              For ilcall1 = 0 To 7
                iMemoriaRAMint(SP + 2, ilcall1) = iBitDPTR(ilcall1)
                iMemoriaRAMint(SP + 1, ilcall1) = iBitDPTR(ilcall1 + 8)
              Next ilcall1
              ActualizacionRAM
              SP = SP + 2
```

```
                    iMemoriaRAMintD(129) = SP
                    ActualizacionRAMD_BH
                End If
            End If
        End If
    End Select
End If
End Sub

Sub Aumento_M1M0_0()
'Esta subrrrutina nos sirve para hacer el aumento
'en el conteo del Timer0
If iMemoriaRAMint(137, 6) = 0 And iMemoriaRAMint(137, 7) = 0 Then
'Timer0 trabaja en el Modo0 temporizador de 13 bits
    iBitDPTR(0) = 0
    iBitDPTR(1) = 0
    iBitDPTR(2) = 0
    For im1m0 = 3 To 10
        iBitDPTR(im1m0) = iMemoriaRAMint(140, im1m0 - 3)
    Next im1m0
    For im1m0 = 11 To 15
        iBitDPTR(im1m0) = iMemoriaRAMint(138, im1m0 - 8)
    Next im1m0
    ConversionDPTRB_D
    im1m0 = iValor16BitsD
    If im1m0 < 8191 Then 'Maximo valor con 13 bits
        If iAumento_Timers Then
            im1m0 = im1m0 + 1
            'Unicamente vamos a añadir cada ciclo de maquina por separado
            'Le añadimos el número de ciclos de máquina
            'de cada instrucción correspondiente
        Else
            If im1m0 <= 8191 - Val(sInstruccionesOperandos(ejec1, 9)) Then
                im1m0 = im1m0 + Val(sInstruccionesOperandos(ejec1, 9))
            Else
                im1m0 = im1m0 - (8191 - Val(sInstruccionesOperandos(ejec1, 9))) - 1
                iMemoriaRAMint(136, 2) = 1 'Activamos bandera TF0
            End If
        End If
        i16Bits1 = im1m0
        ConversionDPTRD_B
        For im1m0 = 0 To 7
            iMemoriaRAMint(140, im1m0) = iBitDPTR(im1m0 + 3)
        Next im1m0
        For im1m0 = 3 To 7
            iMemoriaRAMint(138, im1m0) = iBitDPTR(im1m0 + 8)
        Next im1m0
    Else    'Debemos encerar el Timer y activar la bandera TF0
        For im1m0 = 0 To 7
            iMemoriaRAMint(140, im1m0) = 0
        Next im1m0
        For im1m0 = 3 To 7
            iMemoriaRAMint(138, im1m0) = 0
        Next im1m0
        iMemoriaRAMint(136, 2) = 1 'Activamos bandera TF0
    End If
    ActualizacionRAM
    'Menejo de la etiqueta lblTIMER0
    iBitDPTR(0) = 0
    iBitDPTR(1) = 0
    iBitDPTR(2) = 0
    For im1m0 = 3 To 10
        iBitDPTR(im1m0) = iMemoriaRAMint(140, im1m0 - 3)
    Next im1m0
    For im1m0 = 11 To 15
        iBitDPTR(im1m0) = iMemoriaRAMint(138, im1m0 - 8)
    Next im1m0
    ConversionDPTRB_D
    iX = iValor16BitsD
    ConversionDPTRD_H
    frmDiagrama2.lblTIMER0.Caption = sValor16BitsH & "H"
```

```
    frmDiagrama2.fraTIMER0.Visible = True
    frmDiagrama2.lblTIMER0.Visible = True
  ElseIf iMemoriaRAMint(137, 6) = 0 And iMemoriaRAMint(137, 7) = 1 Then
  'Timer0 trabaja en el Modo1 temporizador/contador de 16 bits
    For im1m0 = 0 To 7
      iBitDPTR(im1m0) = iMemoriaRAMint(140, im1m0)
    Next im1m0
    For im1m0 = 8 To 15
      iBitDPTR(im1m0) = iMemoriaRAMint(138, im1m0 - 8)
    Next im1m0
    ConversionDPTRB_D
    im1m0 = iValor16BitsD
    If im1m0 < 65535 Then 'Maximo valor con 16 bits
      If iAumento_Timers Then
        im1m0 = im1m0 + 1
        'Le añadimos el número de ciclos de máquina
        'de cada instrucción correspondiente
      Else
        If im1m0 <= 65535 - Val(sInstruccionesOperandos(ejec1, 9)) Then
          im1m0 = im1m0 + Val(sInstruccionesOperandos(ejec1, 9))
        Else
          im1m0 = im1m0 - (65535 - Val(sInstruccionesOperandos(ejec1, 9))) - 1
          iMemoriaRAMint(136, 2) = 1 'Activamos bandera TF0
        End If
      End If
      i16Bits1 = im1m0
      ConversionDPTRD_B
      For im1m0 = 0 To 7
        iMemoriaRAMint(140, im1m0) = iBitDPTR(im1m0)
      Next im1m0
      For im1m0 = 0 To 7
        iMemoriaRAMint(138, im1m0) = iBitDPTR(im1m0 + 8)
      Next im1m0
    Else   'Debemos encerar el Timer y activar la bandera TF0
      For im1m0 = 0 To 7
        iMemoriaRAMint(140, im1m0) = 0
        iMemoriaRAMint(138, im1m0) = 0
      Next im1m0
      iMemoriaRAMint(136, 2) = 1 'Activamos bandera TF0
    End If
    ActualizacionRAM
    'Menejo de la etiqueta lblTIMER0
    For im1m0 = 0 To 7
      iBitDPTR(im1m0) = iMemoriaRAMint(140, im1m0)
    Next im1m0
    For im1m0 = 8 To 15
      iBitDPTR(im1m0) = iMemoriaRAMint(138, im1m0 - 8)
    Next im1m0
    ConversionDPTRB_D
    iX = iValor16BitsD
    ConversionDPTRD_H
    frmDiagrama2.lblTIMER0.Caption = sValor16BitsH & "H"
    frmDiagrama2.fraTIMER0.Visible = True
    frmDiagrama2.lblTIMER0.Visible = True
  ElseIf iMemoriaRAMint(137, 6) = 1 And iMemoriaRAMint(137, 7) = 0 Then
  'Timer0 trabaja en el Modo2 temporizador/contador de 8 bits con autorecarga
    For im1m0 = 0 To 7
      iValorB(im1m0) = iMemoriaRAMint(138, im1m0)
    Next im1m0
    ConversionB_D
    im1m0 = iValorD
    If im1m0 < 255 Then 'Maximo valor con 8 bits
      If iAumento_Timers Then
        im1m0 = im1m0 + 1
        'Le añadimos el número de ciclos de máquina
        'de cada instrucción correspondiente
      Else
        If im1m0 <= 255 - Val(sInstruccionesOperandos(ejec1, 9)) Then
          im1m0 = im1m0 + Val(sInstruccionesOperandos(ejec1, 9))
        Else
          im1m0 = im1m0 - (255 - Val(sInstruccionesOperandos(ejec1, 9))) - 1
```

```
            iMemoriaRAMint(136, 2) = 1 'Activamos bandera TF0
          End If
        End If
        iX1d_b = im1m0
        ConversionD_B
        For im1m0 = 0 To 7
          iMemoriaRAMint(138, im1m0) = iMatrizB(im1m0)
        Next im1m0
      Else   'Debemos encerar el Timer y activar la bandera TF0
        For im1m0 = 0 To 7
          'Hacemos autorecarga
          iMemoriaRAMint(138, im1m0) = iMemoriaRAMint(140, im1m0)
        Next im1m0
        iMemoriaRAMint(136, 2) = 1 'Activamos bandera TF0
      End If
      ActualizacionRAM
      'Manejo de etiqueta lblTIMER0
      For im1m0 = 0 To 7
        iValorB(im1m0) = iMemoriaRAMint(138, im1m0)
      Next im1m0
      ConversionB_D
      iX = iValorD
      ConversionD_H
      frmDiagrama2.lblTIMER0.Caption = sValorH & "H"
      frmDiagrama2.fraTIMER0.Visible = True
      frmDiagrama2.lblTIMER0.Visible = True
    Else
    'Timer0 trabaja en el Modo3 contadores multiples específicos
    'dos de 8 bits cada uno
    'Para TH0 dependerá del valor de TR1, por ello no consta aqui
      For im1m0 = 0 To 7
        iValorB(im1m0) = iMemoriaRAMint(138, im1m0)
      Next im1m0
      ConversionB_D
      im1m0 = iValorD
    If im1m0 < 255 Then 'Maximo valor con 8 bits
        If iAumento_Timers Then
          im1m0 = im1m0 + 1
          'Le añadimos el número de ciclos de máquina
          'de cada instrucción correspondiente
        Else
          If im1m0 <= 255 - Val(sInstruccionesOperandos(ejec1, 9)) Then
            im1m0 = im1m0 + Val(sInstruccionesOperandos(ejec1, 9))
          Else
            im1m0 = im1m0 - (255 - Val(sInstruccionesOperandos(ejec1, 9))) - 1
            iMemoriaRAMint(136, 2) = 1 'Activamos bandera TF0
          End If
        End If
        iX1d_b = im1m0
        ConversionD_B
        For im1m0 = 0 To 7
          iMemoriaRAMint(138, im1m0) = iMatrizB(im1m0)
        Next im1m0
      Else   'Debemos encerar el Timer y activar la bandera TF0
        For im1m0 = 0 To 7
          iMemoriaRAMint(138, im1m0) = 0
        Next im1m0
        iMemoriaRAMint(136, 2) = 1 'Activamos bandera TF0
      End If
      ActualizacionRAM
      'Manejo de etiqueta lblTIMER0
      For im1m0 = 0 To 7
        iValorB(im1m0) = iMemoriaRAMint(138, im1m0)
      Next im1m0
      ConversionB_D
      iX = iValorD
      ConversionD_H
      frmDiagrama2.lblTIMER0.Caption = sValorH & "H"
      frmDiagrama2.fraTIMER0.Visible = True
      frmDiagrama2.lblTIMER0.Visible = True
    End If
```

```
End Sub

Sub Aumento_M1M0_1()
'Esta subrrrutina nos sive para hacer el aumento
'en el conteo del Timer1
If iMemoriaRAMint(137, 2) = 0 And iMemoriaRAMint(137, 3) = 0 Then
'Timer1 trabaja en el Modo0 temporizador de 13 bits
   iBitDPTR(0) = 0
   iBitDPTR(1) = 0
   iBitDPTR(2) = 0
   For im1m0 = 3 To 10
      iBitDPTR(im1m0) = iMemoriaRAMint(141, im1m0 - 3)
   Next im1m0
   For im1m0 = 11 To 15
      iBitDPTR(im1m0) = iMemoriaRAMint(139, im1m0 - 8)
   Next im1m0
   ConversionDPTRB_D
   im1m0 = iValor16BitsD
   If im1m0 < 8191 Then 'Maximo valor con 13 bits
      If iAumento_Timers Then
         im1m0 = im1m0 + 1
         'Le añadimos el número de ciclos de máquina
         'de cada instrucción correspondiente
      Else
         If im1m0 <= 8191 - Val(sInstruccionesOperandos(ejec1, 9)) Then
            im1m0 = im1m0 + Val(sInstruccionesOperandos(ejec1, 9))
         Else
            im1m0 = im1m0 - (8191 - Val(sInstruccionesOperandos(ejec1, 9))) - 1
            iMemoriaRAMint(136, 0) = 1 'Activamos bandera TF1
         End If
      End If
      i16Bits1 = im1m0
      ConversionDPTRD_B
      For im1m0 = 0 To 7
         iMemoriaRAMint(141, im1m0) = iBitDPTR(im1m0 + 3)
      Next im1m0
      For im1m0 = 3 To 7
         iMemoriaRAMint(139, im1m0) = iBitDPTR(im1m0 + 8)
      Next im1m0
   Else   'Debemos encerar el Timer y activar la bandera TF1
      For im1m0 = 0 To 7
         iMemoriaRAMint(141, im1m0) = 0
      Next im1m0
      For im1m0 = 3 To 7
         iMemoriaRAMint(139, im1m0) = 0
      Next im1m0
      iMemoriaRAMint(136, 0) = 1 'Activamos.bandera TF1
   End If
   ActualizacionRAM
   'Menejo de la etiqueta lblTIMER1
   iBitDPTR(0) = 0
   iBitDPTR(1) = 0
   iBitDPTR(2) = 0
   For im1m0 = 3 To 10
      iBitDPTR(im1m0) = iMemoriaRAMint(141, im1m0 - 3)
   Next im1m0
   For im1m0 = 11 To 15
      iBitDPTR(im1m0) = iMemoriaRAMint(139, im1m0 - 8)
   Next im1m0
   ConversionDPTRB_D
   iX = iValor16BitsD
   ConversionDPTRD_H
   frmDiagrama2.lblTIMER1.Caption = sValor16BitsH & "H"
   frmDiagrama2.fraTIMER1.Visible = True
   frmDiagrama2.lblTIMER1.Visible = True
ElseIf iMemoriaRAMint(137, 2) = 0 And iMemoriaRAMint(137, 3) = 1 Then
'Timer1 trabaja en el Modo1 temporizador/contador de 16 bits
   For im1m0 = 0 To 7
      iBitDPTR(im1m0) = iMemoriaRAMint(141, im1m0)
   Next im1m0
   For im1m0 = 8 To 15
```

```
        iBitDPTR(im1m0) = iMemoriaRAMint(139, im1m0 - 8)
    Next im1m0
    ConversionDPTRB_D
    im1m0 = iValor16BitsD
    If im1m0 < 65535 Then 'Maximo valor con 16 bits
        If iAumento_Timers Then
            im1m0 = im1m0 + 1
            'Le añadimos el número de ciclos de máquina
            'de cada instrucción correspondiente
        Else
            If im1m0 <= 65535 - Val(sInstruccionesOperandos(ejec1, 9)) Then
                im1m0 = im1m0 + Val(sInstruccionesOperandos(ejec1, 9))
            Else
                im1m0 = im1m0 - (65535 - Val(sInstruccionesOperandos(ejec1, 9))) - 1
                iMemoriaRAMint(136, 0) = 1 'Activamos bandera TF1
            End If
        End If
        i16Bits1 = im1m0
        ConversionDPTRD_B
        For im1m0 = 0 To 7
            iMemoriaRAMint(141, im1m0) = iBitDPTR(im1m0)
        Next im1m0
        For im1m0 = 0 To 7
            iMemoriaRAMint(139, im1m0) = iBitDPTR(im1m0 + 8)
        Next im1m0
    Else    'Debemos encerar el Timer y activar la bandera TF1
        For im1m0 = 0 To 7
            iMemoriaRAMint(141, im1m0) = 0
            iMemoriaRAMint(139, im1m0) = 0
        Next im1m0
        iMemoriaRAMint(136, 0) = 1 'Activamos bandera TF1
    End If
    ActualizacionRAM
    'Menejo de la etiqueta lblTIMER1
    For im1m0 = 0 To 7
        iBitDPTR(im1m0) = iMemoriaRAMint(141, im1m0)
    Next im1m0
    For im1m0 = 8 To 15
        iBitDPTR(im1m0) = iMemoriaRAMint(139, im1m0 - 8)
    Next im1m0
    ConversionDPTRB_D
    iX = iValor16BitsD
    ConversionDPTRD_H
    frmDiagrama2.lblTIMER1.Caption = sValor16BitsH & "H"
    frmDiagrama2.fraTIMER1.Visible = True
    frmDiagrama2.lblTIMER1.Visible = True
ElseIf iMemoriaRAMint(137, 2) = 1 And iMemoriaRAMint(137, 3) = 0 Then
'Timer1 trabaja en el Modo2 temporizador/contador de 8 bits con autorecarga
    For im1m0 = 0 To 7
        iValorB(im1m0) = iMemoriaRAMint(139, im1m0)
    Next im1m0
    ConversionB_D
    im1m0 = iValorD
    If im1m0 < 255 Then 'Maximo valor con 8 bits
        If iAumento_Timers Then
            im1m0 = im1m0 + 1
            'Le añadimos el número de ciclos de máquina
            'de cada instrucción correspondiente
        Else
            If im1m0 <= 255 - Val(sInstruccionesOperandos(ejec1, 9)) Then
                im1m0 = im1m0 + Val(sInstruccionesOperandos(ejec1, 9))
            Else
                im1m0 = im1m0 - (255 - Val(sInstruccionesOperandos(ejec1, 9))) - 1
                iMemoriaRAMint(136, 0) = 1 'Activamos bandera TF1
            End If
        End If
        iX1d_b = im1m0
        ConversionD_B
        For im1m0 = 0 To 7
            iMemoriaRAMint(139, im1m0) = iMatrizB(im1m0)
        Next im1m0
```

```
      Else   'Debemos encerar el Timer y activar la bandera TF1
        For im1m0 = 0 To 7
          'Hacemos autorecarga
          iMemoriaRAMint(139, im1m0) = iMemoriaRAMint(141, im1m0)
        Next im1m0
        iMemoriaRAMint(136, 0) = 1 'Activamos bandera TF1
      End If
      ActualizacionRAM
      'Manejo de etiqueta lblTIMER1
      For im1m0 = 0 To 7
        iValorB(im1m0) = iMemoriaRAMint(139, im1m0)
      Next im1m0
      ConversionB_D
      iX = iValorD
      ConversionD_H
      frmDiagrama2.lblTIMER1.Caption = sValorH & "H"
      frmDiagrama2.fraTIMER1.Visible = True
      frmDiagrama2.lblTIMER1.Visible = True
    ElseIf iMemoriaRAMint(137, 6) = 1 And iMemoriaRAMint(137, 7) = 1 Then
    'Timer0 trabaja en el Modo3 contadores multiples específicos
    'dos de 8 bits cada uno
    'Para TH0 dependerá
      For im1m0 = 0 To 7
        iValorB(im1m0) = iMemoriaRAMint(140, im1m0)
      Next im1m0
      ConversionB_D
      im1m0 = iValorD
      If im1m0 < 255 Then 'Maximo valor con 8 bits
        If iAumento_Timers Then
          im1m0 = im1m0 + 1
          'Le añadimos el número de ciclos de máquina
          'de cada instrucción correspondiente
        Else
          If im1m0 <= 255 - Val(sInstruccionesOperandos(ejec1, 9)) Then
            im1m0 = im1m0 + Val(sInstruccionesOperandos(ejec1, 9))
          Else
            im1m0 = im1m0 - (255 - Val(sInstruccionesOperandos(ejec1, 9))) - 1
            iMemoriaRAMint(136, 0) = 1 'Activamos bandera TF1
          End If
        End If
        iX1d_b = im1m0
        ConversionD_B
        For im1m0 = 0 To 7
          iMemoriaRAMint(140, im1m0) = iMatrizB(im1m0)
        Next im1m0
      Else   'Debemos encerar el Timer y activar la bandera TF1
        For im1m0 = 0 To 7
          iMemoriaRAMint(140, im1m0) = 0
        Next im1m0
        iMemoriaRAMint(136, 0) = 1 'Activamos bandera TF1
      End If
      ActualizacionRAM
      'Manejo de etiqueta lblTIMER1
      For im1m0 = 0 To 7
        iValorB(im1m0) = iMemoriaRAMint(140, im1m0)
      Next im1m0
      ConversionB_D
      iX = iValorD
      ConversionD_H
      frmDiagrama2.lblTIMER1.Caption = sValorH & "H"
      frmDiagrama2.fraTIMER1.Visible = True
      frmDiagrama2.lblTIMER1.Visible = True
    End If
    End Sub

    Sub Byte1_1Ciclo()
    If mnuPorInstruccion.Checked Or mnuTodo.Checked Then
      iAumento_Timers = True
      If mnuBuffer.Checked Then
        frmDiagrama2.Byte1_1Ciclo_1
      Else
```

```
        frmDiagrama1.Byte1_1Ciclo_1
    End If
 ElseIf mnuPorInstruccionSA.Checked Or mnuTodoSA.Checked Then
    iAumento_Timers = False
    Temp_Cont_0
    Temp_Cont_1
    Atencion_Interrupciones
    Exit Sub
 End If
 End Sub

 Sub Byte1_2Ciclo()
 If mnuPorInstruccion.Checked Or mnuTodo.Checked Then
    iAumento_Timers = True
    If mnuBuffer.Checked Then
        frmDiagrama2.Byte1_2Ciclo_1
    Else
        frmDiagrama1.Byte1_2Ciclo_1
    End If
 ElseIf mnuPorInstruccionSA.Checked Or mnuTodoSA.Checked Then
    iAumento_Timers = False
    Temp_Cont_0
    Temp_Cont_1
    Atencion_Interrupciones
    Exit Sub
 End If
 End Sub

 Sub Byte1_2Ciclomovc()
 If mnuPorInstruccion.Checked Or mnuTodo.Checked Then
    iAumento_Timers = True
    If mnuBuffer.Checked Then
        If iRomc1 = 0 Then
            frmDiagrama2.Byte1_2Ciclomovc_1
        Else
            frmDiagrama2.Byte1_2Ciclomovc_NoMOVC
        End If
    Else
        frmDiagrama1.Byte1_2Ciclomovc_1
    End If
 ElseIf mnuPorInstruccionSA.Checked Or mnuTodoSA.Checked Then
    iAumento_Timers = False
    Temp_Cont_0
    Temp_Cont_1
    Atencion_Interrupciones
    Exit Sub
 End If
 End Sub

 Sub Byte1_2CiclomovxR()
 If mnuPorInstruccion.Checked Or mnuTodo.Checked Then
    iAumento_Timers = True
    If mnuBuffer.Checked Then
        Select Case iRamBufferLatch
            Case 1
                frmDiagrama2.Byte1_2CiclomovxR_1
            Case 2
                frmDiagrama2.Byte1_2CiclomovxR_Buffer
            Case 4
                frmDiagrama2.Byte1_2CiclomovxR_NoAdd
        End Select
    Else
        frmDiagrama1.Byte1_2CiclomovxR_1
    End If
 ElseIf mnuPorInstruccionSA.Checked Or mnuTodoSA.Checked Then
    iAumento_Timers = False
    Temp_Cont_0
    Temp_Cont_1
    Atencion_Interrupciones
    Exit Sub
 End If
```

```
End Sub

Sub Byte1_2CiclomovxW()
If mnuPorInstruccion.Checked Or mnuTodo.Checked Then
   iAumento_Timers = True
   If mnuBuffer.Checked Then
      Select Case iRamBufferLatch
         Case 1
            frmDiagrama2.Byte1_2CiclomovxW_1
         Case 3
            frmDiagrama2.Byte1_2CiclomovxW_Latch
         Case 5
            frmDiagrama2.Byte1_2CiclomovxW_NoAdd
      End Select
   Else
      frmDiagrama1.Byte1_2CiclomovxW_1
   End If
ElseIf mnuPorInstruccionSA.Checked Or mnuTodoSA.Checked Then
   iAumento_Timers = False
   Temp_Cont_0
   Temp_Cont_1
   Atencion_Interrupciones
  Exit Sub
End If
End Sub

Sub Byte1_4Ciclo()
If mnuPorInstruccion.Checked Or mnuTodo.Checked Then
   iAumento_Timers = True
   If mnuBuffer.Checked Then
      frmDiagrama2.Byte1_4Ciclo_1
   Else
      frmDiagrama1.Byte1_4Ciclo_1
   End If
ElseIf mnuPorInstruccionSA.Checked Or mnuTodoSA.Checked Then
   iAumento_Timers = False
   Temp_Cont_0
   Temp_Cont_1
   Atencion_Interrupciones
   Exit Sub
End If
End Sub

Sub Byte2_1Ciclo()
If mnuPorInstruccion.Checked Or mnuTodo.Checked Then
   iAumento_Timers = True
   If mnuBuffer.Checked Then
      frmDiagrama2.Byte2_1Ciclo_1
   Else
      frmDiagrama1.Byte2_1Ciclo_1
   End If
ElseIf mnuPorInstruccionSA.Checked Or mnuTodoSA.Checked Then
   iAumento_Timers = False
      frmDiagrama2.EscrituraP1
      If ejec1 > 0 And iEscrituraP1 Then
         If sInstruccionesOperandos(ejec1 - 1, 2) = "144" Or Mid(sInstruccionesOperandos(ejec1 - 1, 2),
                                                                 1, 2) = "P1" Then
            Form2.Puerto_P1WR
         End If
      End If
      frmDiagrama2.LecturaP1
      If sInstruccionesOperandos(ejec1, 3) = "144" Or iLecturaP1 Then
         Form2.Puerto_P1RD
      ElseIf Mid(sInstruccionesOperandos(ejec1, 3), 1, 2) = "P1" _
            Or Mid(sInstruccionesOperandos(ejec1, 3), 1, 3) = "/P1" Or iLecturaP1 Then
         'Para de uno de los 4 pines de entrada en  P1
         Form2.Puerto_P1RD
      End If
   Temp_Cont_0
   Temp_Cont_1
   Atencion_Interrupciones
```

```
    Exit Sub
  End If
  End Sub

  Sub Byte2_2Ciclo()
  If mnuPorInstruccion.Checked Or mnuTodo.Checked Then
    iAumento_Timers = True
    If mnuBuffer.Checked Then
      frmDiagrama2.Byte2_2Ciclo_1
    Else
      frmDiagrama1.Byte2_2Ciclo_1
    End If
  ElseIf mnuPorInstruccionSA.Checked Or mnuTodoSA.Checked Then
    iAumento_Timers = False
      frmDiagrama2.LecturaP1
      If sInstruccionesOperandos(ejec1, 3) = "144" Or iLecturaP1 Then
        Form2.Puerto_P1RD
      ElseIf Mid(sInstruccionesOperandos(ejec1, 3), 1, 2) = "P1" _
          Or Mid(sInstruccionesOperandos(ejec1, 3), 1, 3) = "/P1" Or iLecturaP1 Then
        'Para de uno de los 4 pines de entrada en  P1
        Form2.Puerto_P1RD
      End If
    Temp_Cont_0
    Temp_Cont_1
    Atencion_Interrupciones
    Exit Sub
  End If
  End Sub

  Sub Byte3_2Ciclo()
  If mnuPorInstruccion.Checked Or mnuTodo.Checked Then
    iAumento_Timers = True
    If mnuBuffer.Checked Then
      frmDiagrama2.Byte3_2Ciclo_1
    Else
      frmDiagrama1.Byte3_2Ciclo_1
    End If
  ElseIf mnuPorInstruccionSA.Checked Or mnuTodoSA.Checked Then
    iAumento_Timers = False
      frmDiagrama2.LecturaP1
      If sInstruccionesOperandos(ejec1, 3) = "144" Or iLecturaP1 Then
        Form2.Puerto_P1RD
      ElseIf Mid(sInstruccionesOperandos(ejec1, 3), 1, 2) = "P1" _
          Or Mid(sInstruccionesOperandos(ejec1, 3), 1, 3) = "/P1" Or iLecturaP1 Then
        'Para de uno de los 4 pines de entrada en  P1
        Form2.Puerto_P1RD
      End If
    Temp_Cont_0
    Temp_Cont_1
    Atencion_Interrupciones
    Exit Sub
  End If
  End Sub


  Sub Conv_HexAddP0P2_1()
  'Nos permite convertir el contenido de las variables sAddP2, sAddP0
  'y sDatP0 en caracteres hexadecimales
  LineaLeida1 = sAddP0
  For ip2p0 = 1 To 8
    LetrasLeidas = Mid(LineaLeida1, ip2p0, 1)
    iValorB(ip2p0 - 1) = Val(LetrasLeidas)
  Next ip2p0
  ConversionB_D
  iX = iValorD
  ConversionD_H
  sAddP0 = sValorH
  LineaLeida1 = sDatP0
  For ip2p0 = 1 To 8
    LetrasLeidas = Mid(LineaLeida1, ip2p0, 1)
    iValorB(ip2p0 - 1) = Val(LetrasLeidas)
```

```
Next ip2p0
ConversionB_D
iX = iValorD
ConversionD_H
sDatP0 = sValorH
LineaLeida1 = sAddP2
For ip2p0 = 0 To 3
   iValorB(ip2p0) = 0
Next ip2p0
For ip2p0 = 1 To 4
   LetrasLeidas = Mid(LineaLeida1, ip2p0, 1)
   iValorB(ip2p0 + 3) = Val(LetrasLeidas)
Next ip2p0
ConversionB_D
iX = iValorD
ConversionD_H
sAddP2 = Mid(sValorH, 2, 1)
End Sub

Sub EjecucionAD()
Select Case BitADD
   Case 0
      BitSobra = 0
      iMemoriaRAMint(224, 7 - Add) = 0
   Case 1
      BitSobra = 0
      iMemoriaRAMint(224, 7 - Add) = 1
   Case 2
      BitSobra = 1
      iMemoriaRAMint(224, 7 - Add) = 0
      If Add = 7 Then
         iMemoriaRAMint(208, 0) = 1
      End If
   Case 3
      BitSobra = 1
      iMemoriaRAMint(224, 7 - Add) = 1
      If Add = 7 Then
         iMemoriaRAMint(208, 0) = 1
      End If
End Select
End Sub

Sub EjecucionADD()
Select Case BitADD
   Case 0
      BitSobra = 0
      iMemoriaRAMint(224, 7 - Add) = 0
   Case 1
      BitSobra = 0
      iMemoriaRAMint(224, 7 - Add) = 1
   Case 2
      BitSobra = 1
      iMemoriaRAMint(224, 7 - Add) = 0
      If Add = 3 Then
         iMemoriaRAMint(208, 1) = 1
      ElseIf Add = 7 Then
         iMemoriaRAMint(208, 0) = 1
      End If
   Case 3
      BitSobra = 1
      iMemoriaRAMint(224, 7 - Add) = 1
      If Add = 3 Then
         iMemoriaRAMint(208, 1) = 1
      ElseIf Add = 7 Then
         iMemoriaRAMint(208, 0) = 1
      End If
End Select
End Sub

Sub EjecucionADDC()
Select Case BitADD
```

```
    Case 0
      BitSobra = 0
      iMemoriaRAMint(224, 7 - Add) = 0
      If Add = 7 Then
        iMemoriaRAMint(208, 0) = 0
      End If
    Case 1
      BitSobra = 0
      iMemoriaRAMint(224, 7 - Add) = 1
      If Add = 7 Then
        iMemoriaRAMint(208, 0) = 0
      End If
    Case 2
      BitSobra = 1
      iMemoriaRAMint(224, 7 - Add) = 0
      If Add = 3 Then
        iMemoriaRAMint(208, 1) = 1
      ElseIf Add = 7 Then
        iMemoriaRAMint(208, 0) = 1
      End If
    Case 3
      BitSobra = 1
      iMemoriaRAMint(224, 7 - Add) = 1
      If Add = 3 Then
        iMemoriaRAMint(208, 1) = 1
      ElseIf Add = 7 Then
        iMemoriaRAMint(208, 0) = 1
      End If
End Select
End Sub

Sub EjecucionSUBB()
Select Case BitSUBB
    Case -2
      BitBorrow = 1
      iMemoriaRAMint(224, 7 - Subb) = 0
      If Subb = 7 Then
        iMemoriaRAMint(208, 0) = 1
      ElseIf Subb = 3 Then
        iMemoriaRAMint(208, 1) = 1
      End If
    Case -1
      BitBorrow = 1
      iMemoriaRAMint(224, 7 - Subb) = 1
      If Subb = 7 Then
        iMemoriaRAMint(208, 0) = 1
      ElseIf Subb = 3 Then
        iMemoriaRAMint(208, 1) = 1
      End If
    Case 0
      BitBorrow = 0
      iMemoriaRAMint(224, 7 - Subb) = 0
      If Subb = 7 Then
        iMemoriaRAMint(208, 0) = 0
      ElseIf Subb = 3 Then
        iMemoriaRAMint(208, 1) = 0
      End If
    Case 1
      BitBorrow = 0
      iMemoriaRAMint(224, 7 - Subb) = 1
      If Subb = 7 Then
        iMemoriaRAMint(208, 0) = 0
      ElseIf Subb = 3 Then
        iMemoriaRAMint(208, 1) = 0
      End If
End Select
BitSUBB = 0
End Sub

Sub Frecuencia_Simulación()
If mnu1.Checked Then
```

```
    Frecuencia_Oscilador = 1
  ElseIf mnu2.Checked Then
    Frecuencia_Oscilador = 2
  ElseIf mnu3.Checked Then
    Frecuencia_Oscilador = 3
  ElseIf mnu4.Checked Then
    Frecuencia_Oscilador = 4
  End If
  End Sub

  Sub InstruccionACALL()
  Y = 0
  Byte2_2Ciclo
  For ilcall = 0 To 29
    If sInstruccionesOperandos(ejec1, 2) = sSaltos_Rel(ilcall) Then
      SP1 = ejec1
      ejec1 = iSaltos_Rel(ilcall) - 1
      iFilasSP = iFilasSP + 1
      iMatrizSP(iFilasSP) = SP1
      i16Bits1 = Val(sInstruccionesOperandos(SP1 + 1, 25))
      ConversionDPTRD_B
      For ilcall1 = 0 To 7
        iMemoriaRAMint(SP + 2, ilcall1) = iBitDPTR(ilcall1)
        iMemoriaRAMint(SP + 1, ilcall1) = iBitDPTR(ilcall1 + 8)
      Next ilcall1
      ActualizacionRAM
      SP = SP + 2
      iMemoriaRAMintD(129) = SP
      ActualizacionRAMD_BH
      Exit For
    End If
  Next ilcall
  End Sub

  Sub InstruccionADD()
  Y = 0
  kmov = 0  'Se utiliza cuando se trata de datos
          'del tipo #data y DIRECT
  UbiBancoRnData  'Nos ubicamos en el Banco actual
              'con la variable UbiBanco
  BitSobra = 0
  BitACC1 = iMemoriaRAMint(224, 0)
  iMemoriaRAMint(208, 1) = 0
  iMemoriaRAMint(208, 0) = 0
  If sInstruccionesOperandos(ejec1, 2) = "A" Then
    For iadd = 1 To 10
      If sInstruccionesOperandos(ejec1, 3) = sMatriz_Operando12(iadd) Then
        kmov = 1
        Select Case iadd
          Case 1 To 8
            Byte1_1Ciclo
            For Add = 0 To 7
              BitADD = iMemoriaRAMint(224, 7 - Add) + iMemoriaRAMint(iadd + UbiBanco - 1, 7 -
                                                                    Add) + BitSobra
              EjecucionADD
            Next Add
            If (BitACC1 = 0 And iMemoriaRAMint(iadd + UbiBanco - 1, 0) = 0) And
                  iMemoriaRAMint(224, 0) = 1 Then
              iMemoriaRAMint(208, 5) = 1
            ElseIf (BitACC1 = 1 And iMemoriaRAMint(iadd + UbiBanco - 1, 0) = 1) And
                  iMemoriaRAMint(224, 0) = 0 Then
              iMemoriaRAMint(208, 5) = 1
            Else
              iMemoriaRAMint(208, 5) = 0
            End If
            ActualizacionRAM
            Exit For
          Case 9 To 10
            Byte1_1Ciclo
            For Add = 0 To 7
```

```
            BitADD = iMemoriaRAMint(224, 7 - Add) + iMemoriaRAMint(iMemoriaRAMintD(iadd +
                                                UbiBanco - 9), 7 - Add) + BitSobra
            EjecucionADD
          Next Add
          If (BitACC1 = 0 And iMemoriaRAMint(iMemoriaRAMintD(iadd + UbiBanco - 9), 0) = 0)
                        And iMemoriaRAMint(224, 0) = 1 Then
            iMemoriaRAMint(208, 5) = 1
          ElseIf (BitACC1 = 1 And iMemoriaRAMint(iMemoriaRAMintD(iadd + UbiBanco - 9), 0) =
                                1) And iMemoriaRAMint(224, 0) = 0 Then
            iMemoriaRAMint(208, 5) = 1
          Else
            iMemoriaRAMint(208, 5) = 0
          End If
          ActualizacionRAM
          Exit For
        End Select
      End If
    Next iadd
    If kmov = 0 Then
      LineaLeida1 = sInstruccionesOperandos(ejec1, 3)
      iLongitud1 = Len(LineaLeida1)
      Y = Y + 1
      LetrasLeidas = Mid(LineaLeida1, Y, 1)
      LetrasLeidas = UCase(LetrasLeidas)
      If LetrasLeidas = "#" Then
        Byte2_1Ciclo
        Num_Data
        iX1d_b = iNum_Data
        ConversionD_B
        For Add = 0 To 7
          iNum_DataB(Add) = iMatrizB(Add)
        Next Add
        For Add = 0 To 7
         BitADD = iMemoriaRAMint(224, 7 - Add) + iNum_DataB(7 - Add) + BitSobra
         EjecucionADD
        Next Add
         If (BitACC1 = 0 And iNum_DataB(0) = 0) And iMemoriaRAMint(224, 0) = 1 Then
           iMemoriaRAMint(208, 5) = 1
         ElseIf (BitACC1 = 1 And iNum_DataB(0) = 1) And iMemoriaRAMint(224, 0) = 0 Then
           iMemoriaRAMint(208, 5) = 1
         Else
           iMemoriaRAMint(208, 5) = 0
         End If
         ActualizacionRAM
      Else
        Byte2_1Ciclo
        Y = Y - 1
        Num_Data
        For Add = 0 To 7
         BitADD = iMemoriaRAMint(224, 7 - Add) + iMemoriaRAMint(iNum_Data, 7 - Add) + BitSobra
         EjecucionADD
        Next Add
         If (BitACC1 = 0 And iMemoriaRAMint(iNum_Data, 0) = 0) And iMemoriaRAMint(224, 0) = 1
                                                                        Then
           iMemoriaRAMint(208, 5) = 1
         ElseIf (BitACC1 = 1 And iMemoriaRAMint(iNum_Data, 0) = 1) And iMemoriaRAMint(224, 0) =
                                                                        0 Then
           iMemoriaRAMint(208, 5) = 1
         Else
           iMemoriaRAMint(208, 5) = 0
         End If
         ActualizacionRAM
      End If
    End If
End If
End Sub

Sub InstruccionADDC()
Y = 0
kmov = 0   'Se utiliza cuando se trata de datos
        'del tipo #data y DIRECT
```

```
UbiBancoRnData  'Nos ubicamos en el Banco actual
            'con la variable UbiBanco
BitSobra = 0
BitACC1 = iMemoriaRAMint(224, 0)
iMemoriaRAMint(208, 1) = 0
If sInstruccionesOperandos(ejec1, 2) = "A" Then
   iMemoriaRAMint(224, 7) = iMemoriaRAMint(224, 7) + iMemoriaRAMint(208, 0)
   For iadd = 1 To 10
      If sInstruccionesOperandos(ejec1, 3) = sMatriz_Operando12(iadd) Then
         kmov = 1
         Select Case iadd
            Case 1 To 8
               Byte1_1Ciclo
               For Add = 0 To 7
                  BitADD = iMemoriaRAMint(224, 7 - Add) + iMemoriaRAMint(iadd + UbiBanco - 1, 7 -
                                                                                    Add) + BitSobra
                  EjecucionADDC
               Next Add
               If (BitACC1 = 0 And iMemoriaRAMint(iadd + UbiBanco - 1, 0) = 0) And
                     iMemoriaRAMint(224, 0) = 1 Then
                  iMemoriaRAMint(208, 5) = 1
               ElseIf (BitACC1 = 1 And iMemoriaRAMint(iadd + UbiBanco - 1, 0) = 1) And
                     iMemoriaRAMint(224, 0) = 0 Then
                  iMemoriaRAMint(208, 5) = 1
               Else
                  iMemoriaRAMint(208, 5) = 0
               End If
               ActualizacionRAM
               Exit For
            Case 9 To 10
               Byte1_1Ciclo
               For Add = 0 To 7
                  BitADD = iMemoriaRAMint(224, 7 - Add) + iMemoriaRAMint(iMemoriaRAMintD(iadd +
                                          UbiBanco - 9), 7 - Add) + BitSobra
                  EjecucionADDC
               Next Add
               If (BitACC1 = 0 And iMemoriaRAMint(iMemoriaRAMintD(iadd + UbiBanco - 9), 0) = 0)
                           And iMemoriaRAMint(224, 0) = 1 Then
                  iMemoriaRAMint(208, 5) = 1
               ElseIf (BitACC1 = 1 And iMemoriaRAMint(iMemoriaRAMintD(iadd + UbiBanco - 9), 0) =
                                          1) And iMemoriaRAMint(224, 0) = 0 Then
                  iMemoriaRAMint(208, 5) = 1
               Else
                  iMemoriaRAMint(208, 5) = 0
               End If
               ActualizacionRAM
               Exit For
         End Select
      End If
   Next iadd
   If kmov = 0 Then
      LineaLeida1 = sInstruccionesOperandos(ejec1, 3)
      iLongitud1 = Len(LineaLeida1)
      Y = Y + 1
      LetrasLeidas = Mid(LineaLeida1, Y, 1)
      LetrasLeidas = UCase(LetrasLeidas)
      If LetrasLeidas = "#" Then
         Byte2_1Ciclo
         Num_Data
         iX1d_b = iNum_Data
         ConversionD_B
         For Add = 0 To 7
            iNum_DataB(Add) = iMatrizB(Add)
         Next Add
         For Add = 0 To 7
            BitADD = iMemoriaRAMint(224, 7 - Add) + iNum_DataB(7 - Add) + BitSobra
            EjecucionADDC
         Next Add
         If (BitACC1 = 0 And iNum_DataB(0) = 0) And iMemoriaRAMint(224, 0) = 1 Then
            iMemoriaRAMint(208, 5) = 1
         ElseIf (BitACC1 = 1 And iNum_DataB(0) = 1) And iMemoriaRAMint(224, 0) = 0 Then
```

```
            iMemoriaRAMint(208, 5) = 1
          Else
            iMemoriaRAMint(208, 5) = 0
          End If
          ActualizacionRAM
        Else
          Byte2_1Ciclo
          Y = Y - 1
          Num_Data
          For Add = 0 To 7
           BitADD = iMemoriaRAMint(224, 7 - Add) + iMemoriaRAMint(iNum_Data, 7 - Add) + BitSobra
           EjecucionADDC
          Next Add
          If (BitACC1 = 0 And iMemoriaRAMint(iNum_Data, 0) = 0) And iMemoriaRAMint(224, 0) = 1
                                                                                    Then
            iMemoriaRAMint(208, 5) = 1
          ElseIf (BitACC1 = 1 And iMemoriaRAMint(iNum_Data, 0) = 1) And iMemoriaRAMint(224, 0) =
                                                                                    0 Then
            iMemoriaRAMint(208, 5) = 1
          Else
            iMemoriaRAMint(208, 5) = 0
          End If
          ActualizacionRAM
        End If
      End If
    End If
End Sub

Sub InstruccionAJMP()
Byte2_2Ciclo
Y = 0
For iajmp = 0 To 29
    If sInstruccionesOperandos(ejec1, 2) = sSaltos_Rel(iajmp) Then
        ejec1 = iSaltos_Rel(iajmp) - 1
        Exit For
    End If
Next iajmp
End Sub

Sub InstruccionANL()
Y = 0
kmov = 0   'Se utiliza cuando se trata de datos
         'del tipo #data y DIRECT
kmov1 = 0  'Se utiliza cuando se trata de datos
         'del tipo #data y DIRECT para el ciclo de iadd1
kbit = 0   ' Para realizar las instrucciones con bits
UbiBancoRnData
If sInstruccionesOperandos(ejec1, 2) = "A" Then
    kbit = 1
    kmov1 = 1
    For iadd = 1 To 10
        If sInstruccionesOperandos(ejec1, 3) = sMatriz_Operando12(iadd) Then
            kmov = 1
            Select Case iadd
                Case 1 To 8
                    Byte1_1Ciclo
                    iMemoriaRAMintD(224) = iMemoriaRAMintD(224) And iMemoriaRAMintD(iadd +
                                                                    UbiBanco - 1)
                    ActualizacionRAMD_BH
                    Exit For
                Case 9 To 10
                    Byte1_1Ciclo
                    iMemoriaRAMintD(224) = iMemoriaRAMintD(224) And
                iMemoriaRAMintD(iMemoriaRAMintD(iadd + UbiBanco - 9))
                    ActualizacionRAMD_BH
                    Exit For
            End Select
        End If
    Next iadd
    If kmov = 0 Then
        LineaLeida1 = sInstruccionesOperandos(ejec1, 3)
```

```
        iLongitud1 = Len(LineaLeida1)
        Y = Y + 1
        LetrasLeidas = Mid(LineaLeida1, Y, 1)
        LetrasLeidas = UCase(LetrasLeidas)
        If LetrasLeidas = "#" Then
          Byte2_1Ciclo
          Num_Data
          iMemoriaRAMintD(224) = iMemoriaRAMintD(224) And iNum_Data
          ActualizacionRAMD_BH
        Else
          Byte2_1Ciclo
          Y = Y - 1
          Num_Data
          iMemoriaRAMintD(224) = iMemoriaRAMintD(224) And iMemoriaRAMintD(iNum_Data)
          ActualizacionRAMD_BH
        End If
      End If
    End If
    'Rutina para trabarjar a nivel de bits
    If kbit = 0 Then
      If sInstruccionesOperandos(ejec1, 2) = "C" Then
        kmov1 = 1
        LineaLeida1 = sInstruccionesOperandos(ejec1, 3)
        iLongitud1 = Len(LineaLeida1)
        LetrasLeidas = Mid(LineaLeida1, 1, 1)
        If LetrasLeidas = "/" Then
          Byte2_2Ciclo
          LetrasLeidas = Mid(LineaLeida1, 2, iLongitud1)
          sInstruccionesOperandos(ejec1, 3) = LetrasLeidas
          iOperando = 3
          Ubicacion_Bit
          iMemoriaRAMint(208, 0) = iMemoriaRAMint(208, 0) And (Not (iMemoriaRAMint(iNum_Byte,
                                                                                    iNum_Bit)))
          sInstruccionesOperandos(ejec1, 3) = "/" & sInstruccionesOperandos(ejec1, 3)
          ActualizacionRAM
        Else
          Byte2_2Ciclo
          iOperando = 3
          Ubicacion_Bit
          iMemoriaRAMint(208, 0) = iMemoriaRAMint(208, 0) And iMemoriaRAMint(iNum_Byte,
                                                                                iNum_Bit)
          ActualizacionRAM
        End If
      End If
    End If
    'Fin de la rutina de bits
    If kmov1 = 0 Then
      LineaLeida1 = sInstruccionesOperandos(ejec1, 2)
      iLongitud1 = Len(LineaLeida1)
      Num_Data
      iNum_Data1 = iNum_Data
      If sInstruccionesOperandos(ejec1, 3) = "A" Then
        Byte2_1Ciclo
        kmov = 1
        iMemoriaRAMintD(iNum_Data1) = iMemoriaRAMintD(iNum_Data1) And
                    iMemoriaRAMintD(224)
        ActualizacionRAMD_BH
      End If
      If kmov = 0 Then
        LineaLeida1 = sInstruccionesOperandos(ejec1, 3)
        iLongitud1 = Len(LineaLeida1)
        Y = 0
        Y = Y + 1
        LetrasLeidas = Mid(LineaLeida1, Y, 1)
        LetrasLeidas = UCase(LetrasLeidas)
        If LetrasLeidas = "#" Then
          Byte3_2Ciclo
          Num_Data
          iMemoriaRAMintD(iNum_Data1) = iMemoriaRAMintD(iNum_Data1) And iNum_Data
          ActualizacionRAMD_BH
        End If
```

```
    End If
  End If
  End Sub

  Sub InstruccionCJNE()
  Y = 0
  UbiBancoRnData  'Nos ubicamos en el Banco actual
              'con la variable UbiBanco
  For iadd1 = 0 To 10
     If sInstruccionesOperandos(ejec1, 2) = sMatriz_Operando12(iadd1) Then
     Select Case iadd1
        Case 0
           LineaLeida1 = sInstruccionesOperandos(ejec1, 3)
           iLongitud1 = Len(LineaLeida1)
           Y = Y + 1
           LetrasLeidas = Mid(LineaLeida1, Y, 1)
           LetrasLeidas = UCase(LetrasLeidas)
           If LetrasLeidas = "#" Then
              Byte3_2Ciclo
              Num_Data
              If iMemoriaRAMintD(224) <> iNum_Data Then
                 For icjne = 0 To 29
                    If sInstruccionesOperandos(ejec1, 4) = sSaltos_Rel(icjne) Then
                       ejec1 = iSaltos_Rel(icjne) - 1
                       Exit For
                    End If
                 Next icjne
              End If
              If iMemoriaRAMintD(224) < iNum_Data Then
                 iMemoriaRAMint(208, 0) = 1
              Else
                 iMemoriaRAMint(208, 0) = 0
              End If
              ActualizacionRAM
           Else
              Y = Y - 1
              Byte3_2Ciclo
              Num_Data
              If iMemoriaRAMintD(224) <> iMemoriaRAMintD(iNum_Data) Then
                 For icjne = 0 To 29
                    If sInstruccionesOperandos(ejec1, 4) = sSaltos_Rel(icjne) Then
                       ejec1 = iSaltos_Rel(icjne) - 1
                       Exit For
                    End If
                 Next icjne
              End If
              If iMemoriaRAMintD(224) < iMemoriaRAMintD(iNum_Data) Then
                 iMemoriaRAMint(208, 0) = 1
              Else
                 iMemoriaRAMint(208, 0) = 0
              End If
              ActualizacionRAM
           End If
           Exit For   'iadd1
        Case 1 To 8
           LineaLeida1 = sInstruccionesOperandos(ejec1, 3)
           iLongitud1 = Len(LineaLeida1)
           Y = Y + 1
           LetrasLeidas = Mid(LineaLeida1, Y, 1)
           LetrasLeidas = UCase(LetrasLeidas)
           If LetrasLeidas = "#" Then
              Byte3_2Ciclo
              Num_Data
              If iMemoriaRAMintD(iadd1 + UbiBanco - 1) <> iNum_Data Then
                 For icjne = 0 To 29
                    If sInstruccionesOperandos(ejec1, 4) = sSaltos_Rel(icjne) Then
                       ejec1 = iSaltos_Rel(icjne) - 1
                    Exit For
                    End If
                 Next icjne
              End If
```

```
            If iMemoriaRAMintD(iadd1 + UbiBanco - 1) < iNum_Data Then
               iMemoriaRAMint(208, 0) = 1
            Else
               iMemoriaRAMint(208, 0) = 0
            End If
            ActualizacionRAM
         End If
         Exit For   'iadd1
      Case 9 To 10
         LineaLeida1 = sInstruccionesOperandos(ejec1, 3)
         iLongitud1 = Len(LineaLeida1)
         Y = Y + 1
         LetrasLeidas = Mid(LineaLeida1, Y, 1)
         LetrasLeidas = UCase(LetrasLeidas)
         If LetrasLeidas = "#" Then
            Byte3_2Ciclo
            Num_Data
            If iMemoriaRAMintD(iMemoriaRAMintD(iadd1 + UbiBanco - 9)) <> iNum_Data Then
               For icjne = 0 To 29
                  If sInstruccionesOperandos(ejec1, 4) = sSaltos_Rel(icjne) Then
                     ejec1 = iSaltos_Rel(icjne) - 1
                  Exit For
                  End If
               Next icjne
            End If
            If iMemoriaRAMintD(iMemoriaRAMintD(iadd1 + UbiBanco - 9)) < iNum_Data Then
               iMemoriaRAMint(208, 0) = 1
            Else
               iMemoriaRAMint(208, 0) = 0
            End If
            ActualizacionRAM
         End If
         Exit For   'iadd1
   End Select
   End If
Next iadd1
End Sub

Sub InstruccionCLR()
Y = 0
If sInstruccionesOperandos(ejec1, 2) = "A" Then
   Byte1_1Ciclo
   For clr = 0 To 7
      iMemoriaRAMint(224, clr) = 0
   Next clr
   ActualizacionRAM
ElseIf sInstruccionesOperandos(ejec1, 2) = "C" Then
   Byte1_1Ciclo
   iMemoriaRAMint(208, 0) = 0
   ActualizacionRAM
Else
   Byte2_1Ciclo
   iOperando = 2
   Ubicacion_Bit
   iMemoriaRAMint(iNum_Byte, iNum_Bit) = 0
   ActualizacionRAM
End If
End Sub

Sub InstruccionCPL()
Y = 0
If sInstruccionesOperandos(ejec1, 2) = "A" Then
   Byte1_1Ciclo
   For cpl = 0 To 7
      If iMemoriaRAMint(224, cpl) = 1 Then
         iMemoriaRAMint(224, cpl) = 0
      Else
         iMemoriaRAMint(224, cpl) = 1
      End If
   Next cpl
   ActualizacionRAM
```

```
ElseIf sInstruccionesOperandos(ejec1, 2) = "C" Then
   Byte1_1Ciclo
   If iMemoriaRAMint(208, 0) = 1 Then
      iMemoriaRAMint(208, 0) = 0
   Else
      iMemoriaRAMint(208, 0) = 1
   End If
   ActualizacionRAM
Else
   Byte2_1Ciclo
   iOperando = 2
   Ubicacion_Bit
   If iMemoriaRAMint(iNum_Byte, iNum_Bit) = 1 Then
      iMemoriaRAMint(iNum_Byte, iNum_Bit) = 0
   Else
      iMemoriaRAMint(iNum_Byte, iNum_Bit) = 1
   End If
   ActualizacionRAM
End If
End Sub

Sub InstruccionDA() 'Esta subrutina permite realizar
                    'el ajuste decimal del acumulador
Byte1_1Ciclo
Y = 0
ad1 = 0
sumaad1 = 0
BitSobra = 0
For iad = 0 To 3
   iValorB(iad) = 0
   iValorB(iad + 4) = iMemoriaRAMint(224, iad + 4)
Next iad
ConversionB_D
If iValorD > 9 Or iMemoriaRAMint(208, 1) = 1 Then
   sumaad1 = iValorD + 6
   ad1 = 1
End If
If sumaad1 > 15 Then
   iMemoriaRAMint(208, 0) = 1
End If
For iad = 0 To 3
   iValorB(iad) = 0
   iValorB(iad + 4) = iMemoriaRAMint(224, iad)
Next iad
ConversionB_D
If iValorD > 9 Or iMemoriaRAMint(208, 0) = 1 Then
   If ad1 = 1 Then
      iX1d_b = 102
      ConversionD_B
      For Add = 0 To 7
         BitADD = iMemoriaRAMint(224, 7 - Add) + iMatrizB(7 - Add) + BitSobra
         EjecucionAD
      Next Add
      ActualizacionRAM
   Else
      iX1d_b = 96
      ConversionD_B
      For Add = 0 To 7
         BitADD = iMemoriaRAMint(224, 7 - Add) + iMatrizB(7 - Add) + BitSobra
         EjecucionAD
      Next Add
      ActualizacionRAM
   End If
Else
   If ad1 = 1 Then
      iX1d_b = 6
      ConversionD_B
      For Add = 0 To 7
         BitADD = iMemoriaRAMint(224, 7 - Add) + iMatrizB(7 - Add) + BitSobra
         EjecucionAD
      Next Add
```

```
        ActualizacionRAM
      End If
    End If
    End Sub

    Sub InstruccionDEC()
    Y = 0
    kmov = 0  'Se utiliza cuando se trata de datos
            'del tipo #data y DIRECT
    UbiBancoRnData  'Nos ubicamos en el Banco actual
                'con la variable UbiBanco
    For iadd = 0 To 10
      If sInstruccionesOperandos(ejec1, 2) = sMatriz_Operando12(iadd) Then
        kmov = 1
        Select Case iadd
          Case 0
            Byte1_1Ciclo
            If iMemoriaRAMintD(224) > 0 Then
              iMemoriaRAMintD(224) = iMemoriaRAMintD(224) - 1
            Else
              iMemoriaRAMintD(224) = 255
            End If
            ActualizacionRAMD_BH
            Exit For
          Case 1 To 8
            Byte1_1Ciclo
            If iMemoriaRAMintD(iadd + UbiBanco - 1) > 0 Then
              iMemoriaRAMintD(iadd + UbiBanco - 1) = iMemoriaRAMintD(iadd + UbiBanco - 1) - 1
            Else
              iMemoriaRAMintD(iadd + UbiBanco - 1) = 255
            End If
            ActualizacionRAMD_BH
            Exit For
          Case 9 To 10
            Byte1_1Ciclo
            If iMemoriaRAMintD(iMemoriaRAMintD(iadd + UbiBanco - 9)) > 0 Then
              iMemoriaRAMintD(iMemoriaRAMintD(iadd + UbiBanco - 9)) =
            iMemoriaRAMintD(iMemoriaRAMintD(iadd + UbiBanco - 9)) - 1
            Else
              iMemoriaRAMintD(iMemoriaRAMintD(iadd + UbiBanco - 9)) = 255
            End If
            ActualizacionRAMD_BH
            Exit For
        End Select
      End If
    Next iadd
    If kmov = 0 Then
      Byte2_1Ciclo
      LineaLeida1 = sInstruccionesOperandos(ejec1, 2)
      iLongitud1 = Len(LineaLeida1)
      Num_Data
      If iMemoriaRAMintD(iNum_Data) > 0 Then
        iMemoriaRAMintD(iNum_Data) = iMemoriaRAMintD(iNum_Data) - 1
      Else
        iMemoriaRAMintD(iNum_Data) = 255
      End If
      ActualizacionRAMD_BH
    End If
    End Sub

    Sub InstruccionDIV()
    Y = 0
    Byte1_4Ciclo
    If sInstruccionesOperandos(ejec1, 2) = "AB" Then
      If iMemoriaRAMintD(240) <> 0 Then
        iDIV1 = iMemoriaRAMintD(224) 'Archiva el valor del accumulador anterior a
                              'realizar la división para luego poder obtener
                              'el valor que se guaradará en el registro B
        iMemoriaRAMintD(224) = iMemoriaRAMintD(224) \ iMemoriaRAMintD(240)
        iDIV2 = 0  'Se utiliza para guaradar el valor del producto del acumulador(que ya contiene
                    'por el resultado de la divisón entera el registro B
```

```
        For div = 1 To iMemoriaRAMintD(240)
          iDIV2 = iDIV2 + iMemoriaRAMintD(224)
        Next div
        iMemoriaRAMintD(240) = iDIV1 - iDIV2
        ActualizacionRAMD_BH
        iMemoriaRAMint(208, 0) = 0
        iMemoriaRAMint(208, 5) = 0
        ActualizacionRAM
      Else
        'Debemos generar numeros randomicos para
        'para archivarlos en el acumulador y B
        'y en algunos casos borrar la bandera del carry
        iMemoriaRAMint(208, 5) = 1
        ActualizacionRAM
      End If
    End If
    End Sub

    Sub InstruccionDJNZ()
    Y = 0
    kmov = 0   'Se utiliza cuando se trata de datos
          'del tipo #data y DIRECT
    UbiBancoRnData  'Nos ubicamos en el Banco actual
              'con la variable UbiBanco
    For iadd1 = 1 To 8
      If sInstruccionesOperandos(ejec1, 2) = sMatriz_Operando12(iadd1) Then
        Byte2_2Ciclo
        kmov = 1
        If iMemoriaRAMintD(iadd1 + UbiBanco - 1) > 0 Then
          iMemoriaRAMintD(iadd1 + UbiBanco - 1) = iMemoriaRAMintD(iadd1 + UbiBanco - 1) - 1
        Else
          iMemoriaRAMintD(iadd1 + UbiBanco - 1) = 255
        End If
        ActualizacionRAMD_BH
        If iMemoriaRAMintD(iadd1 + UbiBanco - 1) <> 0 Then
          For icjne = 0 To 29
            If sInstruccionesOperandos(ejec1, 3) = sSaltos_Rel(icjne) Then
              ejec1 = iSaltos_Rel(icjne) - 1
              Exit For
            End If
          Next icjne
        End If
      End If
      Exit For   'iadd1
    Next iadd1
    If kmov = 0 Then
      Byte3_2Ciclo
      LineaLeida1 = sInstruccionesOperandos(ejec1, 2)
      iLongitud1 = Len(LineaLeida1)
      Num_Data
      If iMemoriaRAMintD(iNum_Data) > 0 Then
        iMemoriaRAMintD(iNum_Data) = iMemoriaRAMintD(iNum_Data) - 1
      Else
        iMemoriaRAMintD(iNum_Data) = 255
      End If
      ActualizacionRAMD_BH
      If iMemoriaRAMintD(iNum_Data) <> 0 Then
        For icjne = 0 To 29
          If sInstruccionesOperandos(ejec1, 3) = sSaltos_Rel(icjne) Then
            ejec1 = iSaltos_Rel(icjne) - 1
            Exit For
          End If
        Next icjne
      End If
    End If
    End Sub

    Sub InstruccionINC()
    Y = 0
    kmov = 0   'Se utiliza cuando se trata de datos
          'del tipo #data y DIRECT
```

```
UbiBancoRnData  'Nos ubicamos en el Banco actual
            'con la variable UbiBanco
For iadd = 0 To 11
   If sInstruccionesOperandos(ejec1, 2) = sMatriz_Operando12(iadd) Then
      kmov = 1
      Select Case iadd
         Case 0
            Byte1_1Ciclo
            If iMemoriaRAMintD(224) < 255 Then
               iMemoriaRAMintD(224) = iMemoriaRAMintD(224) + 1
            Else
               iMemoriaRAMintD(224) = 0
            End If
               ActualizacionRAMD_BH
            Exit For
         Case 1 To 8
            Byte1_1Ciclo
            If iMemoriaRAMintD(iadd + UbiBanco - 1) < 255 Then
               iMemoriaRAMintD(iadd + UbiBanco - 1) = iMemoriaRAMintD(iadd + UbiBanco - 1) + 1
            Else
               iMemoriaRAMintD(iadd + UbiBanco - 1) = 0
            End If
            ActualizacionRAMD_BH
            Exit For
         Case 9 To 10
            Byte1_1Ciclo
            If iMemoriaRAMintD(iMemoriaRAMintD(iadd + UbiBanco - 9)) < 255 Then
               iMemoriaRAMintD(iMemoriaRAMintD(iadd + UbiBanco - 9)) =
            iMemoriaRAMintD(iMemoriaRAMintD(iadd + UbiBanco - 9)) + 1
            Else
               iMemoriaRAMintD(iMemoriaRAMintD(iadd + UbiBanco - 9)) = 0
            End If
            ActualizacionRAMD_BH
            Exit For
         Case 11
            Byte1_2Ciclo
            If iDPTRRamD < 65234 Then
               iDPTRRamD = iDPTRRamD + 1
            Else
               iDPTRRamD = 0
            End If
            ActualizacionDPTRD_BH
            Exit For
      End Select
   End If
Next iadd
If kmov = 0 Then
   Byte2_1Ciclo
   LineaLeida1 = sInstruccionesOperandos(ejec1, 2)
   iLongitud1 = Len(LineaLeida1)
   Num_Data
   If iMemoriaRAMintD(iNum_Data) < 255 Then
      iMemoriaRAMintD(iNum_Data) = iMemoriaRAMintD(iNum_Data) + 1
   Else
      iMemoriaRAMintD(iNum_Data) = 0
   End If
   ActualizacionRAMD_BH
End If
End Sub

Sub InstruccionJB()
Byte3_2Ciclo
Y = 0
iOperando = 2
Ubicacion_Bit
If iMemoriaRAMint(iNum_Byte, iNum_Bit) = 1 Then
   For ijb = 0 To 29
      If sInstruccionesOperandos(ejec1, 3) = sSaltos_Rel(ijb) Then
         ejec1 = iSaltos_Rel(ijb) - 1
         Exit For
      End If
```

```
    Next ijb
End If
End Sub

Sub InstruccionJBC()
Byte3_2Ciclo
Y = 0
iOperando = 2
Ubicacion_Bit
If iMemoriaRAMint(iNum_Byte, iNum_Bit) = 1 Then
    For ijbc = 0 To 29
        If sInstruccionesOperandos(ejec1, 3) = sSaltos_Rel(ijbc) Then
            ejec1 = iSaltos_Rel(ijbc) - 1
            iMemoriaRAMint(iNum_Byte, iNum_Bit) = 0
            Exit For
        End If
    Next ijbc
End If
End Sub

Sub InstruccionJC()
Byte2_2Ciclo
Y = 0
If iMemoriaRAMint(208, 0) = 1 Then
    For ijc = 0 To 29
        If sInstruccionesOperandos(ejec1, 2) = sSaltos_Rel(ijc) Then
            ejec1 = iSaltos_Rel(ijc) - 1
            Exit For
        End If
    Next ijc
End If
End Sub

Sub InstruccionJMP()
Y = 0
Byte1_2Ciclo
If sInstruccionesOperandos(ejec1, 2) = "@A+DPTR" Then
    ijmp = iMemoriaRAMintD(224) + iDPTRRamD
    ejec1 = ijmp - 1
End If
End Sub

Sub InstruccionJNB()
Byte3_2Ciclo
Y = 0
iOperando = 2
Ubicacion_Bit
If iMemoriaRAMint(iNum_Byte, iNum_Bit) = 0 Then
    For ijnb = 0 To 29
        If sInstruccionesOperandos(ejec1, 3) = sSaltos_Rel(ijnb) Then
            ejec1 = iSaltos_Rel(ijnb) - 1
            Exit For
        End If
    Next ijnb
End If
End Sub

Sub InstruccionJNC()
Byte2_2Ciclo
Y = 0
If iMemoriaRAMint(208, 0) = 0 Then
    For ijnc = 0 To 29
        If sInstruccionesOperandos(ejec1, 2) = sSaltos_Rel(ijnc) Then
            ejec1 = iSaltos_Rel(ijnc) - 1
            Exit For
        End If
    Next ijnc
End If
End Sub

Sub InstruccionJNZ()
```

```
Byte2_2Ciclo
Y = 0
If iMemoriaRAMintD(224) <> 0 Then
   For ijnz = 0 To 29
      If sInstruccionesOperandos(ejec1, 2) = sSaltos_Rel(ijnz) Then
         ejec1 = iSaltos_Rel(ijnz) - 1
         Exit For
      End If
   Next ijnz
End If
End Sub

Sub InstruccionJZ()
Byte2_2Ciclo
Y = 0
If iMemoriaRAMintD(224) = 0 Then
   For ijz = 0 To 29
      If sInstruccionesOperandos(ejec1, 2) = sSaltos_Rel(ijz) Then
         ejec1 = iSaltos_Rel(ijz) - 1
         Exit For
      End If
   Next ijz
End If
End Sub

Sub InstruccionLCALL()
Byte3_2Ciclo
Y = 0
For ilcall = 0 To 29
   If sInstruccionesOperandos(ejec1, 2) = sSaltos_Rel(ilcall) Then
      SP1 = ejec1
      ejec1 = iSaltos_Rel(ilcall) - 1
      iFilasSP = iFilasSP + 1
      iMatrizSP(iFilasSP) = SP1
      i16Bits1 = Val(sInstruccionesOperandos(SP1 + 1, 25))
      ConversionDPTRD_B
      For ilcall1 = 0 To 7
         iMemoriaRAMint(SP + 2, ilcall1) = iBitDPTR(ilcall1)
         iMemoriaRAMint(SP + 1, ilcall1) = iBitDPTR(ilcall1 + 8)
      Next ilcall1
      ActualizacionRAM
      SP = SP + 2
      iMemoriaRAMintD(129) = SP
      ActualizacionRAMD_BH
      Exit For
   End If
Next ilcall
End Sub

Sub InstruccionLJMP()
Byte3_2Ciclo
Y = 0
For iljmp = 0 To 29
   If sInstruccionesOperandos(ejec1, 2) = sSaltos_Rel(iljmp) Then
      ejec1 = iSaltos_Rel(iljmp) - 1
      Exit For
   End If
Next iljmp
End Sub

Sub InstruccionMOV()
Y = 0
kmov = 0  'Se utiliza cuando se trata de datos
        'del tipo #data y DIRECT
kmov1 = 0  'Se utiliza cuando se trata de datos
        'del tipo #data y DIRECT para el ciclo de iadd1
kbit = 0  'Para trabajar con bits
UbiBancoRnData  'Nos ubicamos en el Banco actual
            'con la variable UbiBanco
For iadd1 = 0 To 11
   If sInstruccionesOperandos(ejec1, 2) = sMatriz_Operando12(iadd1) Then
```

```
kbit = 1
kmov1 = 1
Select Case iadd1
  Case 0
    For iadd = 1 To 10
      If sInstruccionesOperandos(ejec1, 3) = sMatriz_Operando12(iadd) Then
        kmov = 1
        Select Case iadd
          Case 1 To 8
            Byte1_1Ciclo
            iMemoriaRAMintD(224) = iMemoriaRAMintD(iadd + UbiBanco - 1)
            ActualizacionRAMD_BH
            Exit For   'iadd
          Case 9 To 10
            Byte1_1Ciclo
            iMemoriaRAMintD(224) = iMemoriaRAMintD(iMemoriaRAMintD(iadd + UbiBanco -
                                                                                  9))
            ActualizacionRAMD_BH
            Exit For   'iadd
        End Select
      End If
    Next iadd
    If kmov = 0 Then
      LineaLeida1 = sInstruccionesOperandos(ejec1, 3)
      iLongitud1 = Len(LineaLeida1)
      Y = Y + 1
      LetrasLeidas = Mid(LineaLeida1, Y, 1)
      LetrasLeidas = UCase(LetrasLeidas)
      If LetrasLeidas = "#" Then
        Byte2_1Ciclo
        Num_Data
        iMemoriaRAMintD(224) = iNum_Data
        ActualizacionRAMD_BH
      Else
        Byte2_1Ciclo
        Y = Y - 1
        Num_Data
        iMemoriaRAMintD(224) = iMemoriaRAMintD(iNum_Data)
        ActualizacionRAMD_BH
      End If
    End If
    Exit For   'iadd1
  Case 1 To 8
    If sInstruccionesOperandos(ejec1, 3) = "A" Then
      Byte1_1Ciclo
      kmov = 1
      iMemoriaRAMintD(iadd1 + UbiBanco - 1) = iMemoriaRAMintD(224)
      ActualizacionRAMD_BH
    End If
    If kmov = 0 Then
      LineaLeida1 = sInstruccionesOperandos(ejec1, 3)
      iLongitud1 = Len(LineaLeida1)
      Y = Y + 1
      LetrasLeidas = Mid(LineaLeida1, Y, 1)
      LetrasLeidas = UCase(LetrasLeidas)
      If LetrasLeidas = "#" Then
        Byte2_1Ciclo
        Num_Data
        iMemoriaRAMintD(iadd1 + UbiBanco - 1) = iNum_Data
        ActualizacionRAMD_BH
      Else
        Y = Y - 1
        Byte2_2Ciclo
        Num_Data
        iMemoriaRAMintD(iadd1 + UbiBanco - 1) = iMemoriaRAMintD(iNum_Data)
        ActualizacionRAMD_BH
      End If
    End If
    Exit For   'iadd1
  Case 9 To 10
    If sInstruccionesOperandos(ejec1, 3) = "A" Then
```

```
            Byte1_1Ciclo
            kmov = 1
            iMemoriaRAMintD(iMemoriaRAMintD(iadd1 + UbiBanco - 9)) = iMemoriaRAMintD(224)
            ActualizacionRAMD_BH
          End If
          If kmov = 0 Then
            LineaLeida1 = sInstruccionesOperandos(ejec1, 3)
            iLongitud1 = Len(LineaLeida1)
            Y = Y + 1
            LetrasLeidas = Mid(LineaLeida1, Y, 1)
            LetrasLeidas = UCase(LetrasLeidas)
            If LetrasLeidas = "#" Then
              Byte2_1Ciclo
              Num_Data
              iMemoriaRAMintD(iMemoriaRAMintD(iadd1 + UbiBanco - 9)) = iNum_Data
              ActualizacionRAMD_BH
            Else
              Byte2_2Ciclo
              Y = Y - 1
              Num_Data
              iMemoriaRAMintD(iMemoriaRAMintD(iadd1 + UbiBanco - 9)) =
                    iMemoriaRAMintD(iNum_Data)
              ActualizacionRAMD_BH
            End If
          End If
          Exit For   'iadd1
        Case 11
          Byte3_2Ciclo
          LineaLeida1 = sInstruccionesOperandos(ejec1, 3)
          Num_Data16
          iDPTRRamD = iNum_Data16
          ActualizacionDPTRD_BH
          Exit For   'iadd1
      End Select
      End If
    Next iadd1
    'Rutina para manejo de Bits
    If kbit = 0 Then
      If sInstruccionesOperandos(ejec1, 2) = "C" Or sInstruccionesOperandos(ejec1, 3) = "C" Then
        kmov1 = 1
        If sInstruccionesOperandos(ejec1, 2) = "C" Then
          Byte2_1Ciclo
          iOperando = 3
          Ubicacion_Bit
          iMemoriaRAMint(208, 0) = iMemoriaRAMint(iNum_Byte, iNum_Bit)
          ActualizacionRAM
        Else
          Byte2_2Ciclo
          iOperando = 2
          Ubicacion_Bit
          iMemoriaRAMint(iNum_Byte, iNum_Bit) = iMemoriaRAMint(208, 0)
          ActualizacionRAM
        End If
      End If
    End If
    'Fin de rutina para bits
    If kmov1 = 0 Then
      LineaLeida1 = sInstruccionesOperandos(ejec1, 2)
      iLongitud1 = Len(LineaLeida1)
      Num_Data
      iNum_Data1 = iNum_Data 'iNum_Data1 es una variable intermedia
                    'para poder trabajar cuando la instruccion
                    'sea del tipo: DIRECT,#data o DIRECT,DIRECT
                    'para el primer operando
      For iadd = 0 To 10
        If sInstruccionesOperandos(ejec1, 3) = sMatriz_Operando12(iadd) Then
          kmov = 1
          Select Case iadd
            Case 0
              Byte2_1Ciclo
              iMemoriaRAMintD(iNum_Data1) = iMemoriaRAMintD(224)
```

```
            ActualizacionRAMD_BH
            Exit For   'iadd
          Case 1 To 8
            Byte2_2Ciclo
            iMemoriaRAMintD(iNum_Data1) = iMemoriaRAMintD(iadd + UbiBanco - 1)
            ActualizacionRAMD_BH
            Exit For   'iadd
          Case 9 To 10
            Byte2_2Ciclo
            iMemoriaRAMintD(iNum_Data1) = iMemoriaRAMintD(iMemoriaRAMintD(iadd +
                                                    UbiBanco - 9))
            ActualizacionRAMD_BH
            Exit For   'iadd
        End Select
      End If
    Next iadd
    If kmov = 0 Then
      Y = 0
      LineaLeida1 = sInstruccionesOperandos(ejec1, 3)
      iLongitud1 = Len(LineaLeida1)
      Y = Y + 1
      LetrasLeidas = Mid(LineaLeida1, Y, 1)
      LetrasLeidas = UCase(LetrasLeidas)
      If LetrasLeidas = "#" Then
        Byte3_2Ciclo
        Num_Data
        iMemoriaRAMintD(iNum_Data1) = iNum_Data
        ActualizacionRAMD_BH
      Else
        Y = Y - 1
        Byte3_2Ciclo
        Num_Data
        iMemoriaRAMintD(iNum_Data1) = iMemoriaRAMintD(iNum_Data)
        ActualizacionRAMD_BH
      End If
    End If
  End If
End Sub

Sub InstruccionMOVC()
Y = 0
sAddP0 = ""
sDatP0 = ""
sAddP2 = ""
If sInstruccionesOperandos(ejec1, 2) = "A" Then
  If sInstruccionesOperandos(ejec1, 3) = "@A+DPTR" Then
    imovc = iMemoriaRAMintD(224) + iDPTRRamD
    i16Bits1 = imovc
    ConversionDPTRD_B
    For iadr1 = 4 To 7
      sAddP2 = sAddP2 & iBitDPTR(iadr1)
    Next iadr1
    For iadr1 = 8 To 15
      sAddP0 = sAddP0 & iBitDPTR(iadr1)
    Next iadr1
    iRomc1 = 0  'Se calcula el valor que dan los bits
            'P2.7, P2.6 y P2.5, P2.4 no interesa
    For iRBL = 0 To 2
      iRomc1 = iRomc1 + iBitDPTR(iRBL) * 2 ^ (2 - iRBL)
    Next iRBL
    If iRomc1 = 0 Then 'Se puede activar la habilitación de lectura de la
                'memoria Rom externa
      Select Case KbytesRom
        Case 1024
          For iRCMOV = 0 To 5
            iBitDPTR(iRCMOV) = 0 'Dejamos solo los últimos 10 bits
                        'de la dirección
          Next iRCMOV
        Case 2048
          For iRCMOV = 0 To 4
            iBitDPTR(iRCMOV) = 0 'Dejamos solo los últimos 11 bits
```

```
                                'de la dirección
            Next iRCMOV
          Case 4096
            For iRCMOV = 0 To 3
              iBitDPTR(iRCMOV) = 0 'Dejamos solo los últimos 12 bits
                                'de la dirección
            Next iRCMOV
        End Select
        ConversionDPTRB_D
        imovc = iValor16BitsD
        For iadr1 = 0 To 7
          sDatP0 = sDatP0 & iMemoriaROMext(imovc, iadr1)
        Next iadr1
      Else
        'No se habilitará la lectura de la Rom externa y el dato leido
        'será indeterminado, ya que la localidad direccionada no es valida
        For iadr1 = 0 To 7
          sDatP0 = sDatP0 & 1
        Next iadr1
      End If
      Conv_HexAddP0P2_1
      Byte1_2Ciclomovc
      If iRomc1 = 0 Then
        For imovc1 = 0 To 7
          iValorB(imovc1) = iMemoriaROMext(imovc, imovc1)
        Next imovc1
        ConversionB_D
      Else
        iValorD = 255
      End If
      iMemoriaRAMintD(224) = iValorD
      ActualizacionRAMD_BH
    ElseIf sInstruccionesOperandos(ejec1, 3) = "@A+PC" Then
      imovc = iMemoriaRAMintD(224) + Val(sInstruccionesOperandos(ejec1 + 1, 25))
      i16Bits1 = imovc
      ConversionDPTRD_B
      For iadr1 = 4 To 7
        sAddP2 = sAddP2 & iBitDPTR(iadr1)
      Next iadr1
      For iadr1 = 8 To 15
        sAddP0 = sAddP0 & iBitDPTR(iadr1)
      Next iadr1
      iRomc1 = 0  'Se calcula el valor que dan los bits
              'P2.7, P2.6 y P2.5, P2.4 no interesa
      For iRBL = 0 To 2
        iRomc1 = iRomc1 + iBitDPTR(iRBL) * 2 ^ (2 - iRBL)
      Next iRBL
      If iRomc1 = 0 Then 'Se puede activar la habilitación de lectura de la
                  'memoria Rom externa
        Select Case KbytesRom
          Case 1024
            For iRCMOV = 0 To 5
              iBitDPTR(iRCMOV) = 0 'Dejamos solo los últimos 10 bits
                                'de la dirección
            Next iRCMOV
          Case 2048
            For iRCMOV = 0 To 4
              iBitDPTR(iRCMOV) = 0 'Dejamos solo los últimos 11 bits
                                'de la dirección
            Next iRCMOV
          Case 4096
            For iRCMOV = 0 To 3
              iBitDPTR(iRCMOV) = 0 'Dejamos solo los últimos 12 bits
                                'de la dirección
            Next iRCMOV
        End Select
        ConversionDPTRB_D
        imovc = iValor16BitsD
        For iadr1 = 0 To 7
          sDatP0 = sDatP0 & iMemoriaROMext(imovc, iadr1)
        Next iadr1
```

```
        Else
          'No se habilitará la lectura de la Rom externa y el dato leido
          'será indeterminado, ya que la localidad direccionada no es valida
          For iadr1 = 0 To 7
            sDatP0 = sDatP0 & 1
          Next iadr1
        End If
        Conv_HexAddP0P2_1
        Byte1_2Ciclomovc
        If iRomc1 = 0 Then
          For imovc1 = 0 To 7
            iValorB(imovc1) = iMemoriaROMext(imovc, imovc1)
          Next imovc1
          ConversionB_D
        Else
          iValorD = 255
        End If
        iMemoriaRAMintD(224) = iValorD
        ActualizacionRAMD_BH
      End If
    End If
    End Sub

    Sub InstruccionMOVX()
    Y = 0
    UbiBancoRnData  'Nos ubicamos en el Banco actual
              'con la variable UbiBanco
    iRamBufferLatch = 1 'Predefinimos como si se tratará de
                'memoria RAM
    p24 = 0
    sAddP0 = ""
    sDatP0 = ""
    sAddP2 = ""
    For iadd1 = 0 To 12
      If sInstruccionesOperandos(ejec1, 2) = sMatriz_Operando12(iadd1) Then
        Select Case iadd1
          Case 0
            For iadd = 9 To 12
              If sInstruccionesOperandos(ejec1, 3) = sMatriz_Operando12(iadd) Then
                Select Case iadd
                  Case 9 To 10
                    sAddP2 = "0000"
                    i16Bits1 = iMemoriaRAMintD(iadd + UbiBanco - 9)
                    ConversionDPTRD_B
                    For iadr1 = 8 To 15
                      sAddP0 = sAddP0 & iBitDPTR(iadr1)
                    Next iadr1
                    For iadr1 = 0 To 7
                      sDatP0 = sDatP0 & iMemoriaRAMext(iMemoriaRAMintD(iadd + UbiBanco - 9),
                                                                                  iadr1)
                    Next iadr1
                    Conv_HexAddP0P2_1
                    iRamBufferLatch1 = 0
                    iRamBufferLatch = 4
                    Byte1_2CiclomovxR
                    For imovx = 0 To 7
                      iMemoriaRAMint(224, imovx) = iMemoriaRAMext(iMemoriaRAMintD(iadd +
                                                  UbiBanco - 9), imovx)
                    Next imovx
                    ActualizacionRAM
                    Exit For    'iadd
                  Case 12
                    i16Bits1 = iDPTRRamD
                    ConversionDPTRD_B
                    For iadr1 = 4 To 7
                      sAddP2 = sAddP2 & iBitDPTR(iadr1)
                    Next iadr1
                    For iadr1 = 8 To 15
                      sAddP0 = sAddP0 & iBitDPTR(iadr1)
                    Next iadr1
                    iRamBufferLatch1 = 0
```

```
For iRBL = 0 To 2
   iRamBufferLatch1 = iRamBufferLatch1 + iBitDPTR(iRBL) * 2 ^ (2 - iRBL)
Next iRBL
Select Case iRamBufferLatch1
   'Case 0 esta reservado para la memoria ROM
   Case 1
      If iRam = 1 Then
         iRamBufferLatch = 1
      ElseIf iBuffer = 1 Then
         iRamBufferLatch = 2
      Else
         iRamBufferLatch = 4 'Se usará si en la decoficación no resulta
                             'niguna dirección valida de lectura de "RAM"
                             'externa
      End If
   Case 2
      If iRam = 2 Then
         iRamBufferLatch = 1
      ElseIf iBuffer = 2 Then
         iRamBufferLatch = 2
      Else
         iRamBufferLatch = 4
      End If
   Case 3
      If iRam = 3 Then
         iRamBufferLatch = 1
      ElseIf iBuffer = 3 Then
         iRamBufferLatch = 2
      Else
         iRamBufferLatch = 4
      End If
   Case 4
      If iRam = 4 Then
         iRamBufferLatch = 1
      ElseIf iBuffer = 4 Then
         iRamBufferLatch = 2
      Else
         iRamBufferLatch = 4
      End If
   Case 5
      If iRam = 5 Then
         iRamBufferLatch = 1
      ElseIf iBuffer = 5 Then
         iRamBufferLatch = 2
      Else
         iRamBufferLatch = 4
      End If
   Case 6
      If iRam = 6 Then
         iRamBufferLatch = 1
      ElseIf iBuffer = 6 Then
         iRamBufferLatch = 2
      Else
         iRamBufferLatch = 4
      End If
   Case 7
      If iRam = 7 Then
         iRamBufferLatch = 1
      ElseIf iBuffer = 7 Then
         iRamBufferLatch = 2
      Else
         iRamBufferLatch = 4
      End If
End Select
If iRamBufferLatch = 1 Then
   p24 = 0
   If iRam > 1 Then
      For iadr2 = 1 To (iRam - 1) * 2
         p24 = p24 + 4096
      Next iadr2
   Else
```

```
                p24 = 0
              End If
              If iBitDPTR(3) = 0 Then 'Si el bit de direccionamiento P2.4 es 0
                p24 = 8192 + p24
              Else   'Si P24 es 1
                p24 = 12288 + p24
              End If
              If KbytesRam = 1024 Then
                    p24 = p24 + 1024 * (iBitDPTR(4) * 2 + iBitDPTR(5))
              ElseIf KbytesRam = 2048 Then
                    p24 = p24 + 2048 * iBitDPTR(4)
              ElseIf KbytesRam = 4096 Then
                p24 = p24
              End If
              For iadr1 = 0 To 7
                sDatP0 = sDatP0 & iMemoriaRAMext(iDPTRRamD - p24, iadr1)
              Next iadr1
            End If
            Conv_HexAddP0P2_1
            Byte1_2CiclomovxR
            If iRamBufferLatch = 2 Then 'Si es el Buffer
              For imovx = 0 To 7
                iMemoriaRAMint(224, imovx) = saValor_InB_Buffer(imovx)
              Next imovx
            ElseIf iRamBufferLatch = 1 Then 'Si es la RAM
              For imovx = 0 To 7
                iMemoriaRAMint(224, imovx) = iMemoriaRAMext(iDPTRRamD - p24, imovx)
              Next imovx
            Else 'No hay una dirección valida
              For imovx = 0 To 7
                iMemoriaRAMint(224, imovx) = 0
              Next imovx
            End If
            ActualizacionRAM
            Exit For   'iadd
        End Select
      End If
    Next iadd
    Exit For   'iadd1
  Case 9 To 10
    If sInstruccionesOperandos(ejec1, 3) = "A" Then
      i16Bits1 = iMemoriaRAMintD(iadd1 + UbiBanco - 9)
      ConversionDPTRD_B
      sAddP2 = "0000"
      For iadr1 = 8 To 15
        sAddP0 = sAddP0 & iBitDPTR(iadr1)
      Next iadr1
      For iadr1 = 0 To 7
        sDatP0 = sDatP0 & iMemoriaRAMint(224, iadr1)
      Next iadr1
      Conv_HexAddP0P2_1
      iRamBufferLatch1 = 0
      iRamBufferLatch = 5
      Byte1_2CiclomovxW
      For imovx = 0 To 7
        iMemoriaRAMext(iMemoriaRAMintD(iadd1 + UbiBanco - 9), imovx) =
              iMemoriaRAMint(224, imovx)
      Next imovx
    End If
    Exit For   'iadd1
  Case 12
    If sInstruccionesOperandos(ejec1, 3) = "A" Then
        i16Bits1 = iDPTRRamD
        ConversionDPTRD_B
        For iadr1 = 4 To 7
          sAddP2 = sAddP2 & iBitDPTR(iadr1)
        Next iadr1
        For iadr1 = 8 To 15
          sAddP0 = sAddP0 & iBitDPTR(iadr1)
        Next iadr1
        For iadr1 = 0 To 7
```

```
    sDatP0 = sDatP0 & iMemoriaRAMint(224, iadr1)
  Next iadr1
  iRamBufferLatch1 = 0
  For iRBL = 0 To 2
    iRamBufferLatch1 = iRamBufferLatch1 + iBitDPTR(iRBL) * 2 ^ (2 - iRBL)
  Next iRBL
  Select Case iRamBufferLatch1
    'Case 0 esta reservado para la memoria ROM
    Case 1
      If iRam = 1 Then
        iRamBufferLatch = 1
      ElseIf iLatch = 1 Then
        iRamBufferLatch = 3
      Else
        iRamBufferLatch = 5 'Se usará si en la decoficación no resulta
                            'niguna dirección valida de escritura de "RAM"
                            'externa
      End If
    Case 2
      If iRam = 2 Then
        iRamBufferLatch = 1
      ElseIf iLatch = 2 Then
        iRamBufferLatch = 3
      Else
        iRamBufferLatch = 5
      End If
    Case 3
      If iRam = 3 Then
        iRamBufferLatch = 1
      ElseIf iLatch = 3 Then
        iRamBufferLatch = 3
      Else
        iRamBufferLatch = 5
      End If
    Case 4
      If iRam = 4 Then
        iRamBufferLatch = 1
      ElseIf iLatch = 4 Then
        iRamBufferLatch = 3
      Else
        iRamBufferLatch = 5
      End If
    Case 5
      If iRam = 5 Then
        iRamBufferLatch = 1
      ElseIf iLatch = 5 Then
        iRamBufferLatch = 3
      Else
        iRamBufferLatch = 5
      End If
    Case 6
      If iRam = 6 Then
        iRamBufferLatch = 1
      ElseIf iLatch = 6 Then
        iRamBufferLatch = 3
      Else
        iRamBufferLatch = 5
      End If
    Case 7
      If iRam = 7 Then
        iRamBufferLatch = 1
      ElseIf iLatch = 7 Then
        iRamBufferLatch = 3
      Else
        iRamBufferLatch = 5
      End If
  End Select
  Conv_HexAddP0P2_1
  Byte1_2CiclomovxW
  If iRamBufferLatch = 3 Then 'Si el Latch
    For imovx = 0 To 7
```

```
                saValor_Latch377(imovx) = iMemoriaRAMint(224, imovx)
              Next imovx
            ElseIf iRamBufferLatch = 1 Then   'Si es la RAM
              p24 = 0
              If iRam > 1 Then
                For iadr2 = 1 To (iRam - 1) * 2
                  p24 = p24 + 4096
                Next iadr2
              Else
                p24 = 0
              End If
              If iBitDPTR(3) = 0 Then 'Si el bit de direccionamiento P2.4 es 0
                p24 = 8192 + p24
              Else    'Si P24 es 1
                p24 = p24 + 12288
              End If
              If KbytesRam = 1024 Then
                p24 = p24 + 1024 * (iBitDPTR(4) * 2 + iBitDPTR(5))
              ElseIf KbytesRam = 2048 Then
                p24 = p24 + 2048 * iBitDPTR(4)
              ElseIf KbytesRam = 4096 Then
                  p24 = p24
              End If
              For imovx = 0 To 7
                iMemoriaRAMext(iDPTRRamD - p24, imovx) = iMemoriaRAMint(224, imovx)
              Next imovx
            Else 'No hay una dirección valida
              'No se direcciona ningun dato
            End If
          End If
          Exit For    'iadd1
      End Select
    End If
  Next iadd1
  End Sub

  Sub InstruccionMUL()
  Y = 0
  Byte1_4Ciclo
  If sInstruccionesOperandos(ejec1, 2) = "AB" Then
    i16Bits1 = 0
    For mul = 1 To iMemoriaRAMintD(224)
      i16Bits1 = i16Bits1 + iMemoriaRAMintD(240)
    Next mul
    If i16Bits1 > 255 Then
      iMemoriaRAMint(208, 5) = 1
    Else
      iMemoriaRAMint(208, 5) = 0
    End If
    iMemoriaRAMint(208, 0) = 0
    ConversionDPTRD_B
    For mul = 0 To 7
      iMemoriaRAMint(240, mul) = iBitDPTR(mul)
    Next mul
    For mul = 0 To 7
      iMemoriaRAMint(224, mul) = iBitDPTR(mul + 8)
    Next mul
    ActualizacionRAM
  End If
  End Sub

  Sub InstruccionNOP()
  'Establecer un Timer para esperar que esta instruccion
  'se realice
  Byte1_1Ciclo
  PauseTime = 5   'Establece la duración de 5 segundos.
  Start = Timer   'Establece la hora de inicio.
  Do While Timer < Start + PauseTime
  Loop
  End Sub
```

```
Sub InstruccionORL()
Y = 0
kmov = 0  'Se utiliza cuando se trata de datos
        'del tipo #data y DIRECT
kmov1 = 0  'Se utiliza cuando se trata de datos
        'del tipo #data y DIRECT para el ciclo de iadd1
kbit = 0  ' Para realizar las instrucciones con bits
UbiBancoRnData
If sInstruccionesOperandos(ejec1, 2) = "A" Then
   kbit = 1
   kmov1 = 1
   For iadd = 1 To 10
      If sInstruccionesOperandos(ejec1, 3) = sMatriz_Operando12(iadd) Then
         kmov = 1
         Select Case iadd
            Case 1 To 8
               Byte1_1Ciclo
               iMemoriaRAMintD(224) = iMemoriaRAMintD(224) Or iMemoriaRAMintD(iadd +
                                                    UbiBanco - 1)
               ActualizacionRAMD_BH
               Exit For
            Case 9 To 10
               Byte1_1Ciclo
               iMemoriaRAMintD(224) = iMemoriaRAMintD(224) Or
            iMemoriaRAMintD(iMemoriaRAMintD(iadd + UbiBanco - 9))
               ActualizacionRAMD_BH
               Exit For
         End Select
      End If
   Next iadd
   If kmov = 0 Then
      LineaLeida1 = sInstruccionesOperandos(ejec1, 3)
      iLongitud1 = Len(LineaLeida1)
      Y = Y + 1
      LetrasLeidas = Mid(LineaLeida1, Y, 1)
      LetrasLeidas = UCase(LetrasLeidas)
      If LetrasLeidas = "#" Then
         Byte2_1Ciclo
         Num_Data
         iMemoriaRAMintD(224) = iMemoriaRAMintD(224) Or iNum_Data
         ActualizacionRAMD_BH
      Else
         Byte2_1Ciclo
         Y = Y - 1
         Num_Data
         iMemoriaRAMintD(224) = iMemoriaRAMintD(224) Or iMemoriaRAMintD(iNum_Data)
         ActualizacionRAMD_BH
      End If
   End If
End If
'Rutina para trabarjar a nivel de bits
If kbit = 0 Then
   If sInstruccionesOperandos(ejec1, 2) = "C" Then
      kmov1 = 1
      LineaLeida1 = sInstruccionesOperandos(ejec1, 3)
      iLongitud1 = Len(LineaLeida1)
      LetrasLeidas = Mid(LineaLeida1, 1, 1)
      If LetrasLeidas = "/" Then
         Byte2_2Ciclo
         LetrasLeidas = Mid(LineaLeida1, 2, iLongitud1)
         sInstruccionesOperandos(ejec1, 3) = LetrasLeidas
         iOperando = 3
         Ubicacion_Bit
         iMemoriaRAMint(208, 0) = iMemoriaRAMint(208, 0) Or (Not (iMemoriaRAMint(iNum_Byte,
                                                    iNum_Bit)))
         sInstruccionesOperandos(ejec1, 3) = "/" & sInstruccionesOperandos(ejec1, 3)
         ActualizacionRAM
      Else
         Byte2_2Ciclo
         iOperando = 3
         Ubicacion_Bit
```

```
            iMemoriaRAMint(208, 0) = iMemoriaRAMint(208, 0) Or iMemoriaRAMint(iNum_Byte,
                                                                    iNum_Bit)
            ActualizacionRAM
         End If
      End If
   End If
   'Fin de la rutina de bits
   If kmov1 = 0 Then
      LineaLeida1 = sInstruccionesOperandos(ejec1, 2)
      iLongitud1 = Len(LineaLeida1)
      Num_Data
      iNum_Data1 = iNum_Data
      If sInstruccionesOperandos(ejec1, 3) = "A" Then
         Byte2_1Ciclo
         kmov = 1
         iMemoriaRAMintD(iNum_Data1) = iMemoriaRAMintD(iNum_Data1) Or iMemoriaRAMintD(224)
         ActualizacionRAMD_BH
      End If
      If kmov = 0 Then
         LineaLeida1 = sInstruccionesOperandos(ejec1, 3)
         iLongitud1 = Len(LineaLeida1)
         Y = 0
         Y = Y + 1
         LetrasLeidas = Mid(LineaLeida1, Y, 1)
         LetrasLeidas = UCase(LetrasLeidas)
         If LetrasLeidas = "#" Then
            Byte3_2Ciclo
            Num_Data
            iMemoriaRAMintD(iNum_Data1) = iMemoriaRAMintD(iNum_Data1) Or iNum_Data
            ActualizacionRAMD_BH
         End If
      End If
   End If
   End Sub

   Sub InstruccionPOP()
   Byte2_2Ciclo
   Y = 0
   If sInstruccionesOperandos(ejec1, 2) = "A" Then
      iMemoriaRAMintD(224) = iMemoriaRAMintD(SP)
   Else
      LineaLeida1 = sInstruccionesOperandos(ejec1, 2)
      iLongitud1 = Len(LineaLeida1)
      Num_Data
      iMemoriaRAMintD(iNum_Data) = iMemoriaRAMintD(SP)
   End If
   SP = SP - 1
   iMemoriaRAMintD(129) = SP   'Almacenamos el Stack Pointer
   ActualizacionRAMD_BH
   End Sub

   Sub InstruccionPUSH()
   Byte2_2Ciclo
   Y = 0
   SP = SP + 1
   If sInstruccionesOperandos(ejec1, 2) = "A" Then
      iMemoriaRAMintD(SP) = iMemoriaRAMintD(224)
   Else
      LineaLeida1 = sInstruccionesOperandos(ejec1, 2)
      iLongitud1 = Len(LineaLeida1)
      Num_Data
      iMemoriaRAMintD(SP) = iMemoriaRAMintD(iNum_Data)
   End If
   iMemoriaRAMintD(129) = SP   'Almacenamos el Stack Pointer
   ActualizacionRAMD_BH
   End Sub

   Sub InstruccionRET()
   Y = 0
   Byte1_2Ciclo
   ejec1 = iMatrizSP(iFilasSP)
```

```
iFilasSP = iFilasSP - 1
SP = SP - 2
iMemoriaRAMintD(129) = SP   'Almacenamos el Stack Pointer
ActualizacionRAMD_BH
End Sub

Sub InstruccionRETI()
Byte1_2Ciclo
Select Case EA 'EA se define en la subrutina Atencion_Interrupciones
   Case 1
      'Si la interrupción es externa 0
      iInterrupcion_EXTI0 = False
   Case 2
      'Si la interrupción es del Timer0
      iInterrupcion_Timer0 = False
   Case 3
      'Si la interrupción es externa 1
      iInterrupcion_EXTI1 = False
   Case 4
      'Si la interrupción es del Timer1
      iInterrupcion_Timer1 = False
End Select
ejec1 = iMatrizSP(iFilasSP)
iFilasSP = iFilasSP - 1
SP = SP - 2
iMemoriaRAMintD(129) = SP   'Almacenamos el Stack Pointer
ActualizacionRAMD_BH
End Sub

Sub InstruccionRL()
Y = 0
Byte1_1Ciclo
If sInstruccionesOperandos(ejec1, 2) = "A" Then
   BitRL = iMemoriaRAMint(224, 0) 'BitRL guarda el mas significativo del
                                  'acumulador momentaneamente
   For rl = 1 To 7
      iMemoriaRAMint(224, rl - 1) = iMemoriaRAMint(224, rl)
   Next rl
   iMemoriaRAMint(224, 7) = BitRL
   ActualizacionRAM
End If
End Sub

Sub InstruccionRLC()
Y = 0
Byte1_1Ciclo
If sInstruccionesOperandos(ejec1, 2) = "A" Then
   BitRLC = iMemoriaRAMint(224, 0) 'BitRLC guarda el mas significativo del
                                   'acumulador momentaneamente
   For rlc = 1 To 7
      iMemoriaRAMint(224, rlc - 1) = iMemoriaRAMint(224, rlc)
   Next rlc
   iMemoriaRAMint(224, 7) = iMemoriaRAMint(208, 0)
   iMemoriaRAMint(208, 0) = BitRLC
   ActualizacionRAM
End If
End Sub

Sub InstruccionRR()
Y = 0
Byte1_1Ciclo
If sInstruccionesOperandos(ejec1, 2) = "A" Then
   BitRR = iMemoriaRAMint(224, 7) 'BitRR guarda el mas significativo del
                                  'acumulador momentaneamente
   rri = 6
   For rr = 0 To 6
      iMemoriaRAMint(224, rri - rr + 1) = iMemoriaRAMint(224, rri - rr)
   Next rr
   iMemoriaRAMint(224, 0) = BitRR
   ActualizacionRAM
End If
```

```
End Sub

Sub InstruccionRRC()
Y = 0
Byte1_1Ciclo
If sInstruccionesOperandos(ejec1, 2) = "A" Then
   BitRRC = iMemoriaRAMint(224, 7) 'BitRRC guarda el mas significativo del
                                   'acumulador momentaneamente
   rrci = 6
   For rrc = 0 To 6
      iMemoriaRAMint(224, rrci - rrc + 1) = iMemoriaRAMint(224, rrci - rrc)
   Next rrc
   iMemoriaRAMint(224, 0) = iMemoriaRAMint(208, 0)
   iMemoriaRAMint(208, 0) = BitRRC
   ActualizacionRAM
End If
End Sub

Sub InstruccionSETB()
Y = 0
kmov = 0
If sInstruccionesOperandos(ejec1, 2) = "C" Then
   Byte1_1Ciclo
   kmov = 1
   iMemoriaRAMint(208, 0) = 1
End If
If kmov = 0 Then
   Byte2_1Ciclo
   iOperando = 2
   Ubicacion_Bit
   iMemoriaRAMint(iNum_Byte, iNum_Bit) = 1
End If
ActualizacionRAM
End Sub

Sub InstruccionSJMP()
Byte2_2Ciclo
kmov = 0
Y = 0
For isjmp = 0 To 29
   If sInstruccionesOperandos(ejec1, 2) = sSaltos_Rel(isjmp) Then
      kmov = 1
      ejec1 = iSaltos_Rel(isjmp) - 1
      Exit For
   End If
Next isjmp
If kmov = 0 And sInstruccionesOperandos(ejec1, 2) = "$" Then
   Ressjmp = MsgBox("Estamos en un lazo Infinito....Desea continuar?", vbOKCancel + vbQuestion, "V.I.M.P")
      If Ressjmp = vbOK Then
         ejec1 = -1
         Step = 0
      End If
End If
End Sub

Sub InstruccionSUBB()
Y = 0
kmov = 0   'Se utiliza cuando se trata de datos
        'del tipo #data y DIRECT
UbiBancoRnData  'Nos ubicamos en el Banco actual
            'con la variable UbiBanco
BitBorrow = 0
BitACC2 = iMemoriaRAMint(224, 0)
If sInstruccionesOperandos(ejec1, 2) = "A" Then
   BitSUBB = iMemoriaRAMint(208, 0) 'Es para poder tomar el valor del carry en la primera
                                    'operacion de resta al acumulador
   For iadd = 1 To 10
      If sInstruccionesOperandos(ejec1, 3) = sMatriz_Operando12(iadd) Then
         kmov = 1
         Select Case iadd
            Case 1 To 8
```

```
            Byte1_1Ciclo
            For Subb = 0 To 7
              BitSUBB = iMemoriaRAMint(224, 7 - Subb) - (iMemoriaRAMint(iadd + UbiBanco - 1, 7 -
                                                Subb) + BitBorrow + BitSUBB)
              EjecucionSUBB
            Next Subb
            If (BitACC2 = 0 And iMemoriaRAMint(iadd + UbiBanco - 1, 0) = 1) And
                          iMemoriaRAMint(224, 0) = 1 Then
              iMemoriaRAMint(208, 5) = 1
            ElseIf (BitACC2 = 1 And iMemoriaRAMint(iadd + UbiBanco - 1, 0) = 0) And
                          iMemoriaRAMint(224, 0) = 0 Then
              iMemoriaRAMint(208, 5) = 1
            Else
              iMemoriaRAMint(208, 5) = 0
            End If
            ActualizacionRAM
            Exit For
          Case 9 To 10
            Byte1_1Ciclo
            For Subb = 0 To 7
              BitSUBB = iMemoriaRAMint(224, 7 - Subb) - (iMemoriaRAMint(iMemoriaRAMintD(iadd + UbiBanco - 9),
7 - Subb) + BitBorrow + BitSUBB)
              EjecucionSUBB
            Next Subb
            If (BitACC2 = 0 And iMemoriaRAMint(iMemoriaRAMintD(iadd + UbiBanco - 9), 0) = 1) And
iMemoriaRAMint(224, 0) = 1 Then
              iMemoriaRAMint(208, 5) = 1
            ElseIf (BitACC2 = 1 And iMemoriaRAMint(iMemoriaRAMintD(iadd + UbiBanco - 9), 0) =
                                0) And iMemoriaRAMint(224, 0) = 0 Then
              iMemoriaRAMint(208, 5) = 1
            Else
              iMemoriaRAMint(208, 5) = 0
            End If
            ActualizacionRAM
            Exit For
        End Select
      End If
    Next Iadd
    If kmov = 0 Then
      LineaLeida1 = sInstruccionesOperandos(ejec1, 3)
      iLongitud1 = Len(LineaLeida1)
      Y = Y + 1
      LetrasLeidas = Mid(LineaLeida1, Y, 1)
      LetrasLeidas = UCase(LetrasLeidas)
      If LetrasLeidas = "#" Then
        Byte2_1Ciclo
        Num_Data
        iX1d_b = iNum_Data
        ConversionD_B
        For Add = 0 To 7
          iNum_DataB(Add) = iMatrizB(Add)
        Next Add
        For Subb = 0 To 7
         BitSUBB = iMemoriaRAMint(224, 7 - Subb) - (iNum_DataB(7 - Subb) + BitBorrow + BitSUBB)
         EjecucionSUBB
        Next Subb
        If (BitACC2 = 0 And iNum_DataB(0) = 1) And iMemoriaRAMint(224, 0) = 1 Then
          iMemoriaRAMint(208, 5) = 1
        ElseIf (BitACC2 = 1 And iNum_DataB(0) = 0) And iMemoriaRAMint(224, 0) = 0 Then
          iMemoriaRAMint(208, 5) = 1
        Else
          iMemoriaRAMint(208, 5) = 0
        End If
        ActualizacionRAM
      Else
        Byte2_1Ciclo
        Y = Y - 1
        Num_Data
        For Subb = 0 To 7
         BitSUBB = iMemoriaRAMint(224, 7 - Subb) - (iMemoriaRAMint(iNum_Data, 7 - Subb) +
                                                BitBorrow + BitSUBB)
```

```
          EjecucionSUBB
        Next Subb
        If (BitACC2 = 0 And iMemoriaRAMint(iNum_Data, 0) = 1) And iMemoriaRAMint(224, 0) = 1
                                                                Then
          iMemoriaRAMint(208, 5) = 1
        ElseIf (BitACC2 = 1 And iMemoriaRAMint(iNum_Data, 0) = 0) And iMemoriaRAMint(224, 0) =
                                                                    0 Then
          iMemoriaRAMint(208, 5) = 1
        Else
          iMemoriaRAMint(208, 5) = 0
        End If        ActualizacionRAM
      End If
  ·End If
End If
End Sub

Sub InstruccionSWAP()
Y = 0
Byte1_1Ciclo
If sInstruccionesOperandos(ejec1, 2) = "A" Then
   For swap = 0 To 3
      BitSWAP = iMemoriaRAMint(224, swap)
      iMemoriaRAMint(224, swap) = iMemoriaRAMint(224, swap + 4)
      iMemoriaRAMint(224, swap + 4) = BitSWAP
   Next swap
   ActualizacionRAM
End If
End Sub

Sub InstruccionXCH()
Y = 0
kmov = 0  'Se utiliza cuando se trata de datos
        'del tipo #data y DIRECT
UbiBancoRnData  'Nos ubicamos en el Banco actual
            'con la variable UbiBanco
If sInstruccionesOperandos(ejec1, 2) = "A" Then
   For iadd = 1 To 10
      If sInstruccionesOperandos(ejec1, 3) = sMatriz_Operando12(iadd) Then
         ByteXCH = iMemoriaRAMintD(224)
         kmov = 1
         Select Case iadd
            Case 1 To 8
               Byte1_1Ciclo
               iMemoriaRAMintD(224) = iMemoriaRAMintD(iadd + UbiBanco - 1)
               iMemoriaRAMintD(iadd + UbiBanco - 1) = ByteXCH
               ActualizacionRAMD_BH
               Exit For
            Case 9 To 10
               Byte1_1Ciclo
               iMemoriaRAMintD(224) = iMemoriaRAMintD(iMemoriaRAMintD(iadd + UbiBanco - 9))
               iMemoriaRAMintD(iMemoriaRAMintD(iadd + UbiBanco - 9)) = ByteXCH
               ActualizacionRAMD_BH
               Exit For
         End Select
      End If
   Next iadd
   If kmov = 0 Then
      Byte2_1Ciclo
      ByteXCH = iMemoriaRAMintD(224)
      LineaLeida1 = sInstruccionesOperandos(ejec1, 3)
      iLongitud1 = Len(LineaLeida1)
      Num_Data
      iMemoriaRAMintD(224) = iMemoriaRAMintD(iNum_Data)
      iMemoriaRAMintD(iNum_Data) = ByteXCH
      ActualizacionRAMD_BH
   End If
End If
End Sub

Sub InstruccionXCHD()
Y = 0
```

```
 UbiBancoRnData  'Nos ubicamos en el Banco actual
            'con la variable UbiBanco
 If sInstruccionesOperandos(ejec1, 2) = "A" Then
    For iadd = 9 To 10
       If sInstruccionesOperandos(ejec1, 3) = sMatriz_Operando12(iadd) Then
          Byte1_1Ciclo
          For xchd = 4 To 7
             BitXCHD = iMemoriaRAMint(224, xchd)
             iMemoriaRAMint(224, xchd) = iMemoriaRAMint(iMemoriaRAMintD(iadd + UbiBanco - 9),
                                                                                    xchd)
             iMemoriaRAMint(iMemoriaRAMintD(iadd + UbiBanco - 9), xchd) = BitXCHD
          Next xchd
          ActualizacionRAM
          Exit For
       End If
    Next iadd
 End If
 End Sub

 Sub InstruccionXRL()
 Y = 0
 kmov = 0   'Se utiliza cuando se trata de datos
        'del tipo #data y DIRECT
 kmov1 = 0  'Se utiliza cuando se trata de datos
        'del tipo #data y DIRECT para el ciclo de iadd1
 UbiBancoRnData
 If sInstruccionesOperandos(ejec1, 2) = "A" Then
    kmov1 = 1
    For iadd = 1 To 10
       If sInstruccionesOperandos(ejec1, 3) = sMatriz_Operando12(iadd) Then
          kmov = 1
          Select Case iadd
             Case 1 To 8
                Byte1_1Ciclo
                iMemoriaRAMintD(224) = iMemoriaRAMintD(224) Xor iMemoriaRAMintD(iadd +
                                                                    UbiBanco - 1)
                ActualizacionRAMD_BH
                Exit For
             Case 9 To 10
                Byte1_1Ciclo
                iMemoriaRAMintD(224) = iMemoriaRAMintD(224) Xor iMemoriaRAMintD(iMemoriaRAMintD(iadd +
UbiBanco - 9))
                ActualizacionRAMD_BH
                Exit For
          End Select
       End If
    Next iadd
    If kmov = 0 Then
       LineaLeida1 = sInstruccionesOperandos(ejec1, 3)
       iLongitud1 = Len(LineaLeida1)
       Y = Y + 1
       LetrasLeidas = Mid(LineaLeida1, Y, 1)
       LetrasLeidas = UCase(LetrasLeidas)
       If LetrasLeidas = "#" Then
          Byte2_1Ciclo
          Num_Data
          iMemoriaRAMintD(224) = iMemoriaRAMintD(224) Xor iNum_Data
          ActualizacionRAMD_BH
       Else
          Byte2_1Ciclo
          Y = Y - 1
          Num_Data
          iMemoriaRAMintD(224) = iMemoriaRAMintD(224) Xor iMemoriaRAMintD(iNum_Data)
          ActualizacionRAMD_BH
       End If
    End If
 End If
 If kmov1 = 0 Then
    LineaLeida1 = sInstruccionesOperandos(ejec1, 2)
    iLongitud1 = Len(LineaLeida1)
    Num_Data
```

```
    iNum_Data1 = iNum_Data
    If sInstruccionesOperandos(ejec1, 3) = "A" Then
        Byte2_1Ciclo
        kmov = 1
        iMemoriaRAMintD(iNum_Data1) = iMemoriaRAMintD(iNum_Data1) Xor
                        iMemoriaRAMintD(224)
        ActualizacionRAMD_BH
    End If
    If kmov = 0 Then
        LineaLeida1 = sInstruccionesOperandos(ejec1, 3)
        iLongitud1 = Len(LineaLeida1)
        Y = 0
        Y = Y + 1
        LetrasLeidas = Mid(LineaLeida1, Y, 1)
        LetrasLeidas = UCase(LetrasLeidas)
        If LetrasLeidas = "#" Then
            Byte3_2Ciclo
            Num_Data
            iMemoriaRAMintD(iNum_Data1) = iMemoriaRAMintD(iNum_Data1) Xor iNum_Data
            ActualizacionRAMD_BH
        End If
    End If
End If
End Sub

Sub Num_Data()
ydata = Y
UbicacionDBH2 = 0
For UbicacionDBH1 = 1 To 10
    Y = Y + 1
    LetrasLeidas = Mid(LineaLeida1, Y, 1)
    LetrasLeidas = UCase(LetrasLeidas)
    If LetrasLeidas <> Chr$(32) And Y <> iLongitud1 + 1 Then
        UbicacionDBH2 = UbicacionDBH2 + 1
    ElseIf LetrasLeidas = Chr$(32) Or Y = iLongitud1 + 1 Then
        Exit For
    End If
Next UbicacionDBH1
Select Case UbicacionDBH2
    Case 1
        Y = ydata + 1
        LetrasLeidas = Mid(LineaLeida1, Y, 1)
        LetrasLeidas = UCase(LetrasLeidas)
        iNum_Data = Val(LetrasLeidas)
    Case 2
        Y = ydata + 2
        LetrasLeidas = Mid(LineaLeida1, Y, 1)
        LetrasLeidas = UCase(LetrasLeidas)
        If LetrasLeidas = "H" Or LetrasLeidas = "D" Then
            Y = Y - 2 + 1
            LetrasLeidas = Mid(LineaLeida1, Y, 1)
            LetrasLeidas = UCase(LetrasLeidas)
            iNum_Data = Val(LetrasLeidas)
        Else
            Y = Y - 2 + 1
            LetrasLeidas = Mid(LineaLeida1, Y, 2)
            LetrasLeidas = UCase(LetrasLeidas)
            iNum_Data = Val(LetrasLeidas)
        End If
    Case 3
        Y = ydata + 3
        LetrasLeidas = Mid(LineaLeida1, Y, 1)
        LetrasLeidas = UCase(LetrasLeidas)
        If LetrasLeidas = "H" Then
            Y = Y - 3 + 1
            LetrasLeidas = Mid(LineaLeida1, Y, 2)
            LetrasLeidas = UCase(LetrasLeidas)
            sValorH_D = LetrasLeidas
            ConversionH_D
            iNum_Data = iValorH_D
        ElseIf LetrasLeidas = "D" Then
```

```
        Y = Y - 3 + 1
        LetrasLeidas = Mid(LineaLeida1, Y, 2)
        LetrasLeidas = UCase(LetrasLeidas)
        iNum_Data = Val(LetrasLeidas)
      Else
        Y = Y - 3 + 1
        LetrasLeidas = Mid(LineaLeida1, Y, 3)
        LetrasLeidas = UCase(LetrasLeidas)
        iNum_Data = Val(LetrasLeidas)
      End If
    Case 4
      Y = ydata + 4
      LetrasLeidas = Mid(LineaLeida1, Y, 1)
      LetrasLeidas = UCase(LetrasLeidas)
      If LetrasLeidas = "H" Then
        Y = Y - 4 + 2  'Pues los caracteres hexadecimales solo son 2 aun cuando
                       'por regla en ciertos casos se les antepone un cero
        LetrasLeidas = Mid(LineaLeida1, Y, 2)
        LetrasLeidas = UCase(LetrasLeidas)
        sValorH_D = LetrasLeidas
        ConversionH_D
        iNum_Data = iValorH_D
      ElseIf LetrasLeidas = "D" Then
        Y = Y - 4 + 1
        LetrasLeidas = Mid(LineaLeida1, Y, 3)
        LetrasLeidas = UCase(LetrasLeidas)
        iNum_Data = Val(LetrasLeidas)
      End If
    Case 9
      Y = ydata + 9
      LetrasLeidas = Mid(LineaLeida1, Y, 1)
      LetrasLeidas = UCase(LetrasLeidas)
      If LetrasLeidas = "B" Then
        Y = Y - 9
        For datoB = 0 To 7
          Y = Y + 1
          LetrasLeidas = Mid(LineaLeida1, Y, 1)
          iValorB(datoB) = Val(LetrasLeidas)
        Next datoB
        ConversionB_D
        iNum_Data = iValorD
      End If
  End Select
  Y = 0  'Ten cuidado con esta línea de código, puede afectar
      'en la generación de la matriz o en la ejecución del programa
  End Sub

  Sub Num_Data16()
  ydata = 0
  iLongitud1 = Len(LineaLeida1)
  LineaLeida1 = Mid(LineaLeida1, 2, iLongitud1 - 1) 'Permite eliminar el caracter "#"
  iLongitud1 = Len(LineaLeida1)
  LetrasLeidas = Mid(LineaLeida1, iLongitud1, 1)
  LetrasLeidas = UCase(LetrasLeidas)
  If LetrasLeidas = "H" Then
    ydata = iLongitud1 - 1
    Select Case ydata
      Case 5
        LetrasLeidas = Mid(LineaLeida1, 2, 4)
        sValorH_D = LetrasLeidas
        ConversionDPTRH_D
        iNum_Data16 = iValor16BitsD
      Case 4
        LetrasLeidas = Mid(LineaLeida1, 1, 4)
        sValorH_D = LetrasLeidas
        ConversionDPTRH_D
        iNum_Data16 = iValor16BitsD
      Case 3
        LetrasLeidas = Mid(LineaLeida1, 1, 3)
        sValorH_D = 0 & LetrasLeidas
        ConversionDPTRH_D
```

```
        iNum_Data16 = iValor16BitsD
      Case 2
        LetrasLeidas = Mid(LineaLeida1, 1, 2)
        sValorH_D = 0 & 0 & LetrasLeidas
        ConversionDPTRH_D
        iNum_Data16 = iValor16BitsD
      Case 1
        LetrasLeidas = Mid(LineaLeida1, 1, 1)
        iNum_Data16 = Val(LetrasLeidas)
    End Select
ElseIf LetrasLeidas = "D" Then
    ydata = iLongitud1 - 1
    LetrasLeidas = Mid(LineaLeida1, 1, ydata)
    iNum_Data16 = Val(LetrasLeidas)
Else
    ydata = iLongitud1
    LetrasLeidas = Mid(LineaLeida1, 1, ydata)
    iNum_Data16 = Val(LetrasLeidas)
End If
End Sub

Sub PrioridadInterrupcion()
iValorB(0) = 0
iValorB(1) = 0
iValorB(2) = 0
iValorB(3) = 0
For Pri = 4 To 7
    iValorB(Pri) = iMemoriaRAMint(184, Pri)
Next Pri
ConversionB_D
Select Case iValorD
    Case 0, 1, 3, 7, 15
        INT0P = 1
        Timer0P = 2
        INT1P = 3
        Timer1P = 4
    Case 8
        INT0P = 2
        Timer0P = 3
        INT1P = 4
        Timer1P = 1
    Case 4
        INT0P = 2
        Timer0P = 3
        INT1P = 1
        Timer1P = 4
    Case 12
        INT0P = 3
        Timer0P = 4
        INT1P = 1
        Timer1P = 2
    Case 2
        INT0P = 2
        Timer0P = 1
        INT1P = 3
        Timer1P = 4
    Case 10
        INT0P = 3
        Timer0P = 1
        INT1P = 4
        Timer1P = 2
    Case 6
        INT0P = 3
        Timer0P = 1
        INT1P = 2
        Timer1P = 4
    Case 14
        INT0P = 4
        Timer0P = 1
        INT1P = 2
        Timer1P = 3
```

```
    Case 9
      INT0P = 1
      Timer0P = 3
      INT1P = 4
      Timer1P = 2
    Case 5
      INT0P = 1
      Timer0P = 3
      INT1P = 2
      Timer1P = 4
    Case 13
      INT0P = 1
      Timer0P = 4
      INT1P = 2
      Timer1P = 3
    Case 11
      INT0P = 1
      Timer0P = 2
      INT1P = 4
      Timer1P = 3
End Select
End Sub

Sub PSerial_SCON()
'Empesamos averiguando en que modo esta trabajando el
'puerto
If iMemoriaRAMint(152, 0) = 0 And iMemoriaRAMint(152, 1) = 0 Then
   'If SM0 = 0 and SM1 = 0 Estamos en el Modo 0
ElseIf iMemoriaRAMint(152, 0) = 0 And iMemoriaRAMint(152, 1) = 1 Then
   'If SM0 = 0 and SM1 = 1 Estamos en el Modo 1
ElseIf iMemoriaRAMint(152, 0) = 1 And iMemoriaRAMint(152, 1) = 0 Then
   'If SM0 = 1 and SM1 = 0 Estamos en el Modo 2
ElseIf iMemoriaRAMint(152, 0) = 1 And iMemoriaRAMint(152, 1) = 1 Then
   'If SM0 = 1 and SM1 = 1 Estamos en el Modo 3
End If
End Sub

Sub Puerto_P1RD()
'Hacemos una carga inicial del valor de los interruptores
'en el puerto P1 para luego en la subrrutina de MOV
'se muevan los datos al lugar de memoria requerido
For ip1in = 4 To 7
   iMemoriaRAMint(144, ip1in) = saValor_InB_P1(ip1in)
Next ip1in
ActualizacionRAM
End Sub

Sub Puerto_P1WR()
'Subrrutina para manejar la forma en que el micro
'escribe los datos en el pórtico P1
'son los 4 bits menos significativos
'If sInstruccionesOperandos(ejec1 - 1, 2) = "144" Then
If mnuPorticos.Checked Then
   For ip1 = 0 To 7
      If iMemoriaRAMint(144, ip1) = 0 Then
         frmDiagrama1.imgPuerto_P1(7 - ip1).Picture = frmImagenes.imgP1_01.Picture
      Else
         frmDiagrama1.imgPuerto_P1(7 - ip1).Picture = frmImagenes.imgP1_11.Picture
      End If
   Next ip1
ElseIf mnuBuffer.Checked Then
   For ip1 = 0 To 3
      If iMemoriaRAMint(144, ip1) = 0 Then
         frmDiagrama2.imgPuerto_P1(7 - ip1).Picture = frmImagenes.imgP1_01_out.Picture
      Else
         frmDiagrama2.imgPuerto_P1(7 - ip1).Picture = frmImagenes.imgP1_11_out.Picture
      End If
   Next ip1
End If
End Sub
```

```
Sub SaltosBytesVerdaderos()
Dim A1 As Integer
Dim B1 As Integer
ejec1 = -1
iROM = -1
While ejec1 < iMatriz
   ejec1 = ejec1 + 1
   For ejec2 = 0 To 13
      If sInstruccionesOperandos(ejec1, 1) = sSaltosVerdaderos(ejec2) Then
         Select Case ejec2
            Case 0
               ' InstruccionJB
               sInstruccionesOperandos(ejec1, 7) = ""
               For ijb = 0 To 100
                  If sInstruccionesOperandos(ejec1, 3) = sSaltos_Rel(ijb) Then
                     If sInstruccionesOperandos(iSaltos_Rel(ijb), 1) <> "" Then
                        A1 = Val(sInstruccionesOperandos(iSaltos_Rel(ijb), 25))
                        B1 = Val(sInstruccionesOperandos(ejec1 + 1, 25))
                        If A1 >= B1 Then
                           iX1d_b = A1 - B1
                        Else
                           iX1d_b = 256 - (B1 - A1)
                        End If
                     Else
                        A1 = Val(sInstruccionesOperandos(iSaltos_Rel(ijb) + 1, 25))
                        B1 = Val(sInstruccionesOperandos(ejec1 + 1, 25))
                        If A1 >= B1 Then
                           iX1d_b = A1 - B1
                        Else
                           iX1d_b = 256 - (B1 - A1)
                        End If
                     End If
                     Exit For
                  End If
               Next ijb
               ConversionD_B
               For ijb = 0 To 7
                  sInstruccionesOperandos(ejec1, 7) = sInstruccionesOperandos(ejec1, 7) & iMatrizB(ijb)
               Next ijb
               'jb
               Exit For
            Case 1
               ' InstruccionJC
               sInstruccionesOperandos(ejec1, 6) = ""
               For ijc = 0 To 100
                  If sInstruccionesOperandos(ejec1, 2) = sSaltos_Rel(ijc) Then
                     If sInstruccionesOperandos(iSaltos_Rel(ijc), 1) <> "" Then
                        A1 = Val(sInstruccionesOperandos(iSaltos_Rel(ijc), 25))
                        B1 = Val(sInstruccionesOperandos(ejec1 + 1, 25))
                        If A1 >= B1 Then
                           iX1d_b = A1 - B1
                        Else
                           iX1d_b = 256 - (B1 - A1)
                        End If
                     Else
                        A1 = Val(sInstruccionesOperandos(iSaltos_Rel(ijc) + 1, 25))
                        B1 = Val(sInstruccionesOperandos(ejec1 + 1, 25))
                        If A1 >= B1 Then
                           iX1d_b = A1 - B1
                        Else
                           iX1d_b = 256 - (B1 - A1)
                        End If
                     End If
                     Exit For
                  End If
               Next ijc
               ConversionD_B
               For ijc1 = 0 To 7
                  sInstruccionesOperandos(ejec1, 6) = sInstruccionesOperandos(ejec1, 6) & iMatrizB(ijc1)
               Next ijc1
               'jc
```

```
      Exit For
    Case 2
      ' InstruccionJZ
      sInstruccionesOperandos(ejec1, 6) = ""
      For ijz = 0 To 100
        If sInstruccionesOperandos(ejec1, 2) = sSaltos_Rel(ijz) Then
          If sInstruccionesOperandos(iSaltos_Rel(ijz), 1) <> "" Then
            A1 = Val(sInstruccionesOperandos(iSaltos_Rel(ijz), 25))
            B1 = Val(sInstruccionesOperandos(ejec1 + 1, 25))
            If A1 >= B1 Then
              iX1d_b = A1 - B1
            Else
              iX1d_b = 256 - (B1 - A1)
            End If
          Else
            A1 = Val(sInstruccionesOperandos(iSaltos_Rel(ijz) + 1, 25))
            B1 = Val(sInstruccionesOperandos(ejec1 + 1, 25))
            If A1 >= B1 Then
              iX1d_b = A1 - B1
            Else
              iX1d_b = 256 - (B1 - A1)
            End If
          End If
          Exit For
        End If
      Next ijz
      ConversionD_B
      For ijz = 0 To 7
        sInstruccionesOperandos(ejec1, 6) = sInstruccionesOperandos(ejec1, 6) & iMatrizB(ijz)
      Next ijz
      'jz
      Exit For
    Case 3
      ' InstruccionJBC
      sInstruccionesOperandos(ejec1, 7) = ""
      For ijnb = 0 To 29
        If sInstruccionesOperandos(ejec1, 3) = sSaltos_Rel(ijnb) Then
          If sInstruccionesOperandos(iSaltos_Rel(ijnb), 1) <> "" Then
            A1 = Val(sInstruccionesOperandos(iSaltos_Rel(ijnb), 25))
            B1 = Val(sInstruccionesOperandos(ejec1 + 1, 25))
            If A1 >= B1 Then
              ijnb1 = A1 - B1
            Else
              ijnb1 = 256 - (B1 - A1)
            End If
          Else
            A1 = Val(sInstruccionesOperandos(iSaltos_Rel(ijnb) + 1, 25))
            B1 = Val(sInstruccionesOperandos(ejec1 + 1, 25))
            If A1 >= B1 Then
              ijnb1 = A1 - B1
            Else
              ijnb1 = 256 - (B1 - A1)
            End If
          End If
          Exit For
        End If
      Next ijnb
      iX1d_b = ijnb1
      ConversionD_B
      For Add = 0 To 7
        sInstruccionesOperandos(ejec1, 7) = sInstruccionesOperandos(ejec1, 7) & iMatrizB(Add)
      Next Add
      'jbc
      Exit For
    Case 4
      ' InstruccionJNB
      sInstruccionesOperandos(ejec1, 7) = ""
      For ijnb = 0 To 29
        If sInstruccionesOperandos(ejec1, 3) = sSaltos_Rel(ijnb) Then
          If sInstruccionesOperandos(iSaltos_Rel(ijnb), 1) <> "" Then
            A1 = Val(sInstruccionesOperandos(iSaltos_Rel(ijnb), 25))
```

```
            B1 = Val(sInstruccionesOperandos(ejec1 + 1, 25))
            If A1 >= B1 Then
              ijnb1 = A1 - B1
            Else
              ijnb1 = 256 - (B1 - A1)
            End If
          Else
            A1 = Val(sInstruccionesOperandos(iSaltos_Rel(ijnb) + 1, 25))
            B1 = Val(sInstruccionesOperandos(ejec1 + 1, 25))
            If A1 >= B1 Then
              ijnb1 = A1 - B1
            Else
              ijnb1 = 256 - (B1 - A1)
            End If
          End If
          Exit For
        End If
      Next ijnb
      iX1d_b = ijnb1
      ConversionD_B
      For Add = 0 To 7
        sInstruccionesOperandos(ejec1, 7) = sInstruccionesOperandos(ejec1, 7) & iMatrizB(Add)
      Next Add
      'jnb
      Exit For
    Case 5
      ' InstruccionJNC
      sInstruccionesOperandos(ejec1, 6) = ""
      For ijnc = 0 To 29
        If sInstruccionesOperandos(ejec1, 2) = sSaltos_Rel(ijnc) Then
          If sInstruccionesOperandos(iSaltos_Rel(ijnc), 1) <> "" Then
            A1 = Val(sInstruccionesOperandos(iSaltos_Rel(ijnc), 25))
            B1 = Val(sInstruccionesOperandos(ejec1 + 1, 25))
            If A1 >= B1 Then
              ijnc1 = A1 - B1
            Else
              ijnc1 = 256 - (B1 - A1)
            End If
          Else
            A1 = Val(sInstruccionesOperandos(iSaltos_Rel(ijnc) + 1, 25))
            B1 = Val(sInstruccionesOperandos(ejec1 + 1, 25))
            If A1 >= B1 Then
              ijnc1 = A1 - B1
            Else
              ijnc1 = 256 - (B1 - A1)
            End If
          End If
          Exit For
        End If
      Next ijnc
      iX1d_b = ijnc1
      ConversionD_B
      For Add = 0 To 7
        sInstruccionesOperandos(ejec1, 6) = sInstruccionesOperandos(ejec1, 6) & iMatrizB(Add)
      Next Add
      'jnc
      Exit For
    Case 6
      ' InstruccionJNZ
      sInstruccionesOperandos(ejec1, 6) = ""
      For ijnz = 0 To 29
        If sInstruccionesOperandos(ejec1, 2) = sSaltos_Rel(ijnz) Then
          If sInstruccionesOperandos(iSaltos_Rel(ijnz), 1) <> "" Then
            A1 = Val(sInstruccionesOperandos(iSaltos_Rel(ijnz), 25))
            B1 = Val(sInstruccionesOperandos(ejec1 + 1, 25))
            If A1 >= B1 Then
              ijnz1 = A1 - B1
            Else
              ijnz1 = 256 - (B1 - A1)
            End If
          Else
```

```
            A1 = Val(sInstruccionesOperandos(iSaltos_Rel(ijnz) + 1, 25))
            B1 = Val(sInstruccionesOperandos(ejec1 + 1, 25))
            If A1 >= B1 Then
               ijnz1 = A1 - B1                    Else
               ijnz1 = 256 - (B1 - A1)
            End If
         End If
         Exit For
      End If
   Next ijnz
   iX1d_b = ijnz1
   ConversionD_B
   For Add = 0 To 7
      sInstruccionesOperandos(ejec1, 6) = sInstruccionesOperandos(ejec1, 6) & iMatrizB(Add)
   Next Add
   'jnz
   Exit For
Case 7
   ' InstruccionAJMP
   sInstruccionesOperandos(ejec1, 6) = ""
   For iajmp = 0 To 29
      If sInstruccionesOperandos(ejec1, 2) = sSaltos_Rel(iajmp) Then
         If sInstruccionesOperandos(iSaltos_Rel(iajmp), 1) <> "" Then
            iajmp1 = Val(sInstruccionesOperandos(iSaltos_Rel(iajmp), 25))
         Else
            iajmp1 = Val(sInstruccionesOperandos(iSaltos_Rel(iajmp) + 1, 25))
         End If
         Exit For
      End If
   Next iajmp
   i16Bits1 = iajmp1
   ConversionDPTRD_B
   sInstruccionesOperandos(ejec1, 5) = iBitDPTR(5) & iBitDPTR(6) & iBitDPTR(7) &
                                                                  "00001"
   For Add = 0 To 7
      sInstruccionesOperandos(ejec1, 6) = sInstruccionesOperandos(ejec1, 6) & iBitDPTR(Add
                                                                              + 8)
   Next Add
   'ajmp
   Exit For
Case 8
   ' InstruccionCJNE
   sInstruccionesOperandos(ejec1, 7) = ""
   For icjne = 0 To 29
      If sInstruccionesOperandos(ejec1, 4) = sSaltos_Rel(icjne) Then
         If sInstruccionesOperandos(iSaltos_Rel(icjne), 1) <> "" Then
            A1 = Val(sInstruccionesOperandos(iSaltos_Rel(icjne), 25))
            B1 = Val(sInstruccionesOperandos(ejec1 + 1, 25))
            If A1 >= B1 Then
               icjne1 = A1 - B1
            Else
               icjne1 = 256 - (B1 - A1)
            End If
         Else
            A1 = Val(sInstruccionesOperandos(iSaltos_Rel(icjne) + 1, 25))
            B1 = Val(sInstruccionesOperandos(ejec1 + 1, 25))
            If A1 >= B1 Then
               icjne1 = A1 - B1
            Else
               icjne1 = 256 - (B1 - A1)
            End If
         End If
         Exit For
      End If
   Next icjne
   iX1d_b = icjne1
   ConversionD_B
   For Add = 0 To 7
      sInstruccionesOperandos(ejec1, 7) = sInstruccionesOperandos(ejec1, 7) & iMatrizB(Add)
   Next Add
   Exit For
```

```
Case 9
  ' InstruccionDJNZ
  For iadd1 = 1 To 8
    If sInstruccionesOperandos(ejec1, 2) = sMatriz_Operando12(iadd1) Then
      kmov = 1
      sInstruccionesOperandos(ejec1, 6) = ""
      For icjne = 0 To 29
        If sInstruccionesOperandos(ejec1, 3) = sSaltos_Rel(icjne) Then
          If sInstruccionesOperandos(iSaltos_Rel(icjne), 1) <> "" Then
            A1 = Val(sInstruccionesOperandos(iSaltos_Rel(icjne), 25))
            B1 = Val(sInstruccionesOperandos(ejec1 + 1, 25))
            If A1 >= B1 Then
              icjne1 = A1 - B1
            Else
              icjne1 = 256 - (B1 - A1)
            End If
          Else
            A1 = Val(sInstruccionesOperandos(iSaltos_Rel(icjne) + 1, 25))
            B1 = Val(sInstruccionesOperandos(ejec1 + 1, 25))
            If A1 >= B1 Then
              icjne1 = A1 - B1
            Else
              icjne1 = 256 - (B1 - A1)
            End If
          End If
          Exit For
        End If
      Next icjne
      iX1d_b = icjne1
      ConversionD_B
      For Add = 0 To 7
        sInstruccionesOperandos(ejec1, 6) = sInstruccionesOperandos(ejec1, 6) &
                                            iMatrizB(Add)
      Next Add
    End If
    Exit For   'iadd1
  Next iadd1
  If kmov = 0 Then
    sInstruccionesOperandos(ejec1, 7) = ""
    For icjne = 0 To 29
      If sInstruccionesOperandos(ejec1, 3) = sSaltos_Rel(icjne) Then
        If sInstruccionesOperandos(iSaltos_Rel(icjne), 1) <> "" Then
          A1 = Val(sInstruccionesOperandos(iSaltos_Rel(icjne), 25))
          B1 = Val(sInstruccionesOperandos(ejec1 + 1, 25))
          If A1 >= B1 Then
            icjne1 = A1 - B1
          Else
            icjne1 = 256 - (B1 - A1)
          End If
        Else
          A1 = Val(sInstruccionesOperandos(iSaltos_Rel(icjne) + 1, 25))
          B1 = Val(sInstruccionesOperandos(ejec1 + 1, 25))
          If A1 >= B1 Then
            icjne1 = A1 - B1
          Else
            icjne1 = 256 - (B1 - A1)
          End If
        End If
        Exit For
      End If
    Next icjne
    iX1d_b = icjne1
    ConversionD_B
    For Add = 0 To 7
      sInstruccionesOperandos(ejec1, 7) = sInstruccionesOperandos(ejec1, 7) &
                                          iMatrizB(Add)
    Next Add
  End If
  'djnz
  Exit For
Case 10
```

```
    ' InstruccionLJMP
    sInstruccionesOperandos(ejec1, 6) = ""
    sInstruccionesOperandos(ejec1, 7) = ""
    For iljmp = 0 To 29
      If sInstruccionesOperandos(ejec1, 2) = sSaltos_Rel(iljmp) Then
        If sInstruccionesOperandos(iSaltos_Rel(iljmp), 1) <> "" Then
          iljmp1 = Val(sInstruccionesOperandos(iSaltos_Rel(iljmp), 25))
        Else
          iljmp1 = Val(sInstruccionesOperandos(iSaltos_Rel(iljmp) + 1, 25))
        End If
        Exit For
      End If
    Next iljmp
    i16Bits1 = iljmp1
    ConversionDPTRD_B
    For Add = 0 To 7
        sInstruccionesOperandos(ejec1, 6) = sInstruccionesOperandos(ejec1, 6) & iBitDPTR(Add)
sInstruccionesOperandos(ejec1, 7) = sInstruccionesOperandos(ejec1, 7) & iBitDPTR(Add + 8)
    Next Add
    'ljmp
    Exit For
  Case 11
    ' InstruccionSJMP
    sInstruccionesOperandos(ejec1, 6) = ""
    For isjmp = 0 To 29
      If sInstruccionesOperandos(ejec1, 2) = sSaltos_Rel(isjmp) Then
        If sInstruccionesOperandos(iSaltos_Rel(isjmp), 1) <> "" Then
          A1 = Val(sInstruccionesOperandos(iSaltos_Rel(isjmp), 25))
          B1 = Val(sInstruccionesOperandos(ejec1 + 1, 25))
          If A1 >= B1 Then
            isjmp1 = A1 - B1
          Else
            isjmp1 = 256 - (B1 - A1)
          End If
        Else
          A1 = Val(sInstruccionesOperandos(iSaltos_Rel(isjmp) + 1, 25))
          B1 = Val(sInstruccionesOperandos(ejec1 + 1, 25))
          If A1 >= B1 Then
            isjmp1 = A1 - B1
          Else
            isjmp1 = 256 - (B1 - A1)
          End If
        End If
        Exit For
      End If
    Next isjmp
    If sInstruccionesOperandos(ejec1, 2) = "$" Then
      iX1d_b = 254
    Else
      iX1d_b = isjmp1
    End If
    ConversionD_B
    For Add = 0 To 7
      sInstruccionesOperandos(ejec1, 6) = sInstruccionesOperandos(ejec1, 6) & iMatrizB(Add)
    Next Add
    'sjmp
    Exit For
  Case 12
    ' InstruccionACALL
    sInstruccionesOperandos(ejec1, 6) = ""
    For ilcall = 0 To 29
      If sInstruccionesOperandos(ejec1, 2) = sSaltos_Rel(ilcall) Then
        If sInstruccionesOperandos(iSaltos_Rel(ilcall), 1) <> "" Then
          ilcall1 = Val(sInstruccionesOperandos(iSaltos_Rel(ilcall), 25))
        Else
          ilcall1 = Val(sInstruccionesOperandos(iSaltos_Rel(ilcall) + 1, 25))
        End If
        i16Bits1 = ilcall1
        ConversionDPTRD_B
        sInstruccionesOperandos(ejec1, 5) = iBitDPTR(5) & iBitDPTR(6) & iBitDPTR(7) &
                                                                        "10001"
```

```
                For Add = 0 To 7
                  sInstruccionesOperandos(ejec1, 6) = sInstruccionesOperandos(ejec1, 6) &
                                                      iBitDPTR(Add + 8)
                Next Add
                Exit For
              End If
            Next ilcall
            'acall
            Exit For
          Case 13
            ' InstruccionLCALL
            sInstruccionesOperandos(ejec1, 6) = ""
            sInstruccionesOperandos(ejec1, 7) = ""
            For ilcall = 0 To 29
              If sInstruccionesOperandos(ejec1, 2) = sSaltos_Rel(ilcall) Then
                If sInstruccionesOperandos(iSaltos_Rel(ilcall), 1) <> "" Then
                  ilcall1 = Val(sInstruccionesOperandos(iSaltos_Rel(ilcall), 25))
                Else
                  ilcall1 = Val(sInstruccionesOperandos(iSaltos_Rel(ilcall) + 1, 25))
                End If
                i16Bits1 = ilcall1
                ConversionDPTRD_B
                For Add = 0 To 7
                  sInstruccionesOperandos(ejec1, 6) = sInstruccionesOperandos(ejec1, 6) &
                                                      iBitDPTR(Add)
                  sInstruccionesOperandos(ejec1, 7) = sInstruccionesOperandos(ejec1, 7) &
                                                      iBitDPTR(Add + 8)
                Next Add
                Exit For
              End If
            Next ilcall
            'lcall
            Exit For
        End Select
      End If
    Next ejec2
  'Carga de la memoria ROM externa
  ip2p0 = -1
  For iROM1 = 1 To Val(sInstruccionesOperandos(ejec1, 8))
    LineaLeida1 = sInstruccionesOperandos(ejec1, iROM1 + 4)
    iROM = iROM + 1
    ip2p0 = ip2p0 + 1
    For iROM2 = 1 To 8
      LetrasLeidas = Mid(LineaLeida1, iROM2, 1)
      iMemoriaROMext(iROM, iROM2 - 1) = Val(LetrasLeidas)
    Next iROM2
    LineaLeida1 = ""
    'Generamos direccionamiento para P2 y P0, en las
    'columnas 10 y 11 de la matriz sInstruccionesOperandos(n,11)
    'y en las columnas 12,13 y 14,15 dependiendo del número de
    'bytes de la instrucción
      i16Bits1 = iROM
      ConversionDPTRD_B
      For iP2P01 = 4 To 7
        sInstruccionesOperandos(ejec1, 10 + ip2p0 * 2) = sInstruccionesOperandos(ejec1, 10 + ip2p0 *
                                                      2) & iBitDPTR(iP2P01)
      Next iP2P01
      For iP2P01 = 8 To 15
        sInstruccionesOperandos(ejec1, 11 + ip2p0 * 2) = sInstruccionesOperandos(ejec1, 11 + ip2p0 *
                                                      2) & iBitDPTR(iP2P01)
      Next iP2P01
Next iROM1
Wend
End Sub

Sub Smnu1()
mnu1_Click
End Sub

Sub Smnu1KbytesRAM()
mnu1KbytesRAM_Click
```

```
End Sub

Sub Smnu1KbytesROM()
mnu1KbytesROM_Click
End Sub

Sub Smnu2()
mnu2_Click
End Sub

Sub Smnu2KbytesRAM()
mnu2KbytesRAM_Click
End Sub

Sub Smnu2KbytesROM()
mnu2KbytesROM_Click
End Sub

Sub Smnu3()
mnu3_Click
End Sub

Sub Smnu4()
mnu4_Click
End Sub

Sub Smnu4KbytesRAM()
mnu4KbytesRAM_Click
End Sub

Sub Smnu4KbytesROM()
mnu4KbytesROM_Click
End Sub

Sub SmnuAbrir()
mnuAbrir_Click
End Sub

Sub SmnuCodigo()
mnuCodigo_Click
End Sub

Sub SmnuEditarInstruccion()
mnuEditarInstrución_Click
End Sub

Sub SmnuPorInstruccion()
mnuPorInstruccion_Click
End Sub

Sub SmnuPorInstruccionSA()
mnuPorInstruccionSA_Click
End Sub

Sub SmnuRAMInterna()
mnuRAMIntema_Click
End Sub

Sub SmnuROM()
mnuROM_Click
End Sub

Sub SmnuTodo()
mnuTodo_Click
End Sub

Sub SmnuTodoSA()
mnuTodoSA_Click
End Sub

Sub SmnuVer_EsquematicoRam()
```

```
mnuVer_EsquematicoRam_Click
End Sub

Sub SmnuVer_EsquematicoRom()
mnuVer_EsquematicoROM_Click
End Sub

Sub Temp_Cont_0()
'Timer0
If iMemoriaRAMint(136, 3) = 1 Then  'Para revisar si el temporizador
                                    'o contador arranco  TR0
   If iMemoriaRAMint(137, 4) = 0 Then 'Avilita el control por
                                      'software GATE
      If iMemoriaRAMint(137, 5) = 0 Then  'Averiguamos que trabaja como temporizador
                                          'C/T
         Aumento_M1M0_0
      Else   'Entonces es contador y dependera del pin T0
         If ETimer0 Then
            iAumento_Timers = True
            Aumento_M1M0_0
         End If
      End If
   Else   'El control se maneja por Hardware del pin INT0
         'GATE
         'Esta opción no es controlada por el programa
         'ya que Visual Basic no es MultiReading
   End If
Else
   frmDiagrama2.fraTIMER0.Visible = False
   frmDiagrama2.lblTIMER0.Visible = False
End If
End Sub

Sub Temp_Cont_1()
'Timer1
If iMemoriaRAMint(136, 1) = 1 Then  'Para revisar si el temporizador
                                    'o contador arranco  TR1
   If iMemoriaRAMint(137, 0) = 0 Then 'Avilita el control por
                                      'software GATE
      If iMemoriaRAMint(137, 1) = 0 Then  'Averiguamos que trabaja como temporizador
                                          'C/T
         Aumento_M1M0_1
      Else   'Entonces es contador y dependera del pin T1
         If ETimer1 Then
            iAumento_Timers = True
            Aumento_M1M0_1
         End If
      End If
   Else   'El control se maneja por Hardware del pin INT1
         'GATE
         'Esta opción no es controlada por el programa
         'ya que Visual Basic no es MultiReading
   End If
Else
   frmDiagrama2.fraTIMER1.Visible = False
   frmDiagrama2.lblTIMER1.Visible = False
End If
End Sub

Sub UbiBancoRnData()
If iMemoriaRAMint(208, 3) = 0 And iMemoriaRAMint(208, 4) = 0 Then
'Si es el Banco O
   UbiBanco = 0
ElseIf iMemoriaRAMint(208, 3) = 0 And iMemoriaRAMint(208, 4) = 1 Then
'Si es el Banco 1
   UbiBanco = 8
ElseIf iMemoriaRAMint(208, 3) = 1 And iMemoriaRAMint(208, 4) = 0 Then
'Si es el Banco 2
   UbiBanco = 16
ElseIf iMemoriaRAMint(208, 3) = 1 And iMemoriaRAMint(208, 4) = 1 Then
'Si es el Banco 3
```

```
    UbiBanco = 24
  End If
  End Sub

  Sub Ubicacion_Bit()
  'Esta subrutina permite ubicar el byte y al bit dentro del byte
  'con las variables iNum_Byte e iNum_Bit
  kmov = 0
  kmov1 = 0
  For ubi = 0 To 55
    If sInstruccionesOperandos(ejec1, iOperando) = sSFR_Bits(ubi, 0) Then
      kmov = 1
      LineaLeida1 = sSFR_Bits(ubi, 1)
      iLongitud1 = Len(LineaLeida1)
      LetrasLeidas = Mid(LineaLeida1, 1, iLongitud1 - 2)
      iNum_Byte = Val(LetrasLeidas)
      LetrasLeidas = Mid(LineaLeida1, iLongitud1, 1)
      iNum_Bit = 7 - Val(LetrasLeidas)
      Exit For
    End If
  Next ubi
  If kmov = 0 Then
    LineaLeida1 = sInstruccionesOperandos(ejec1, iOperando)
    iLongitud1 = Len(LineaLeida1)
    LetrasLeidas = Mid(LineaLeida1, 1, iLongitud1 - 2)
    For ubi = 0 To 19
      If LetrasLeidas = sSFR_Operandos(ubi, 0) Then
        kmov1 = 1
        iNum_Byte = Val(sSFR_Operandos(ubi, 1))
        LetrasLeidas = Mid(LineaLeida1, iLongitud1, 1)
        iNum_Bit = 7 - Val(LetrasLeidas)
        Exit For
      End If
    Next ubi
    If kmov1 = 0 Then
      iNum_Bit = 7 - Val(Mid(LineaLeida1, iLongitud1, 1))
      LineaLeida1 = LetrasLeidas
      iLongitud1 = Len(LineaLeida1)
      Y = 0
      Num_Data
      iNum_Byte = iNum_Data
    End If
  End If
  End Sub

  Private Sub cmdEjecutarPrograma_Click()
  'Código para la ejecución de las instrucciones
  Frecuencia_Simulación
  ejec1 = -1
  While ejec1 < iMatriz
    ejec1 = ejec1 + 1
    PC = ejec1
    For ejec2 = 0 To 43
      If sInstruccionesOperandos(ejec1, 1) = sMatriz_Opcode(ejec2) Then
        Select Case ejec2
          Case 0
            InstruccionDA   'Esta subrrutina permite
                            'realizar el ajuste decimal
                            'del acumulador
            Exit For
          Case 1
            InstruccionJC
            'jc
            Exit For
          Case 2
            InstruccionJB
            'jb
            Exit For
          Case 3
            InstruccionJZ
            'jz
```

```
    Exit For
  Case 4
    InstruccionRL
    'rl
    Exit For
  Case 5
    InstruccionRR
    'rr
    Exit For
  Case 6
    InstruccionADD
    'add
    Exit For
  Case 7
    InstruccionANL
    'anl
    Exit For
  Case 8
    InstruccionCLR
    'clr
    Exit For
  Case 9
    InstruccionCPL
    'cpl
    Exit For
  Case 10
    InstruccionDIV
    'div
    Exit For
  Case 11
    InstruccionDEC
    'dec
    Exit For
  Case 12
    InstruccionINC
    'inc
    Exit For
  Case 13
    InstruccionJNC
    'jnc
    Exit For
  Case 14
    InstruccionJNB
    'jnb
    Exit For
  Case 15
    InstruccionJBC
    'jbc
    Exit For
  Case 16
    InstruccionJMP
    'jmp
    Exit For
  Case 17
    InstruccionJNZ
    'jnz
    Exit For
  Case 18
    InstruccionMUL
    'mul
    Exit For
  Case 19
    InstruccionMOV
    'mov
    Exit For
  Case 20
    InstruccionNOP
    'nop
    Exit For
  Case 21
    InstruccionORL
```

```
    'orl
    Exit For
Case 22
    InstruccionPOP
    'pop
    Exit For
Case 23
    InstruccionRLC
    'rlc
    Exit For
Case 24
    InstruccionRRC
    'rrc
    Exit For
Case 25
    InstruccionRET
    'ret
    Exit For
Case 26
    InstruccionXRL
    'xrl
    Exit For
Case 27
    InstruccionXCH
    'xch
    Exit For
Case 28
    InstruccionADDC
    Exit For
    'addc
Case 29
    InstruccionAJMP
    'ajmp
    Exit For
Case 30
    InstruccionCJNE
    'cjne
    Exit For
Case 31
    InstruccionDJNZ
    'djnz
    Exit For
Case 32
    InstruccionLJMP
    'ljmp
    Exit For
Case 33
    InstruccionMOVC
    'movc
    Exit For
Case 34
    InstruccionMOVX
    'movx
    Exit For
Case 35
    InstruccionPUSH
    'push
    Exit For
Case 36
    InstruccionRETI
    'reti
    Exit For
Case 37
    InstruccionSUBB
    'subb
    Exit For
Case 38
    InstruccionSWAP
    'swap
    Exit For
Case 39
```

```
            InstruccionSETB
            'setb
            Exit For
          Case 40
            InstruccionSJMP
            'sjmp
            Exit For
          Case 41
            InstruccionXCHD
            'xchd
            Exit For
          Case 42
            InstruccionACALL
            'acall
            Exit For
          Case 43
            InstruccionLCALL
            'lcall
            Exit For
        End Select
      End If
    Next ejec2
  Wend
  End Sub

  Private Sub cmdGenerarMatriz_Click()
  For i = 0 To 2000
    For j = 0 To 24
      sInstruccionesOperandos(i, j) = ""
    Next j
  Next i
  'Utilizamos el menu Abrir de Windows
  'para cargar los programas
  If mnuBuffer.Checked Then  ""
    On Error GoTo Cancel1
    CommonDialog1.CancelError = True
    CommonDialog1.Filter = "Cargar Programa (*.asm)|*.asm|Cargar Programa (*.txt)|*.txt|Todos (*.*)|*.*"
    CommonDialog1.FilterIndex = 1
    CommonDialog1.InitDir = "c:\8031"
    CommonDialog1.Action = 1
    sFileName1 = CommonDialog1.filename
    If sFileName1 = "" Then
  Cancel1:
    Exit Sub
    End If
  Else   ""Si es para el Pórtico Serial
    sFileName1 = "C:\8031\ArUsuario\PSerial.asm"
  End If  ""
  FileNumber1 = FreeFile
  Open sFileName1 For Input As #FileNumber1
  iMatriz = -1
  While Not EOF(FileNumber1)
    Line Input #FileNumber1, LineaLeida1
    iMatriz = iMatriz + 1
    iLongitud1 = Len(LineaLeida1)
    Y = 0
    sOperandosMOV = ""
    columna = 0
    Do Until Y > iLongitud1 Or LetrasLeidas = ";"   'La segunda condición se utiliza cuando es una
                                                    'linea solo de comentario
      Y = Y + 1
      LetrasLeidas = Mid(LineaLeida1, Y, 1)
      LetrasLeidas = UCase(LetrasLeidas)
      If LetrasLeidas <> Chr$(32) Then
        If LetrasLeidas <> ";" Then 'El orden de estos dos If es importante para salirce del ciclo
                                    'de lectura cuando existen comentarios después de la instruccion
        sOperandosMOV = LetrasLeidas
        For i = 0 To 15 'El punto final del contador dependerá de la instruccion
                        'que contenga la cadena las larga de caracteres con todo y operandos
          Y = Y + 1
          LetrasLeidas = Mid(LineaLeida1, Y, 1)
```

```
            LetrasLeidas = UCase(LetrasLeidas)
            If LetrasLeidas <> Chr$(32) And Y <> iLongitud1 + 1 And LetrasLeidas <> "," And LetrasLeidas <> ";" Then
               sOperandosMOV = sOperandosMOV & LetrasLeidas
            ElseIf LetrasLeidas = "," Then
               sInstruccionesOperandos(iMatriz, columna) = sOperandosMOV
               columna = columna + 1
               Exit For
             Else
               sInstruccionesOperandos(iMatriz, columna) = sOperandosMOV
               columna = columna + 1
               Exit For
            End If
         Next i
         End If
      End If
   Loop
   LetrasLeidas = ""   'Esta linea se utiliza para perder la condición
                  'de que LetrasLeidas sea igual a ";"
Wend
Close #FileNumber1
'Ordenamiento de la matriz sInstruccionesOperandos
For j = 0 To iMatriz
   For k = 0 To 43
      If sInstruccionesOperandos(j, 0) = sMatriz_Opcode(k) Then
         For i = 0 To 3
            sInstruccionesOperandos(j, 4 - i) = sInstruccionesOperandos(j, 3 - i)
         Next i
         sInstruccionesOperandos(j, 0) = ""
         Exit For
      End If
   Next k
Next j
'Carga de la matriz de etiquetas sEQU_diDa
For j = 0 To iMatriz
   If sInstruccionesOperandos(j, 1) = "EQU" Then
      sEQU_diDa(j, 0) = sInstruccionesOperandos(j, 0)
      sEQU_diDa(j, 1) = sInstruccionesOperandos(j, 2)
   End If
Next j
'Carga de las matrices de saltos sSaltos_Rel y iSaltos_Rel
irel = 0
For j = 0 To iMatriz
   If sInstruccionesOperandos(j, 0) <> "" Then
      If sInstruccionesOperandos(j, 1) <> "EQU" Then
         sSaltos_Rel(irel) = sInstruccionesOperandos(j, 0)
         iSaltos_Rel(irel) = j
         irel = irel + 1
      End If
   End If
Next j
'Actualizacion de la matriz sInstruccionesOperandos en base
'a las matrices sEQU_diDa y sSFR_Operandos
'Para la matriz sEQU_diDa
For j = 0 To iMatriz
   For i = 0 To 29
      If sInstruccionesOperandos(j, 2) = sEQU_diDa(i, 0) Then
         sInstruccionesOperandos(j, 2) = sEQU_diDa(i, 1)
      End If
      If sInstruccionesOperandos(j, 2) = "#" & sEQU_diDa(i, 0) Then
         sInstruccionesOperandos(j, 2) = "#" & sEQU_diDa(i, 1)
      End If
      If sInstruccionesOperandos(j, 3) = sEQU_diDa(i, 0) Then
         sInstruccionesOperandos(j, 3) = sEQU_diDa(i, 1)
      End If
      If sInstruccionesOperandos(j, 3) = "#" & sEQU_diDa(i, 0) Then
         sInstruccionesOperandos(j, 3) = "#" & sEQU_diDa(i, 1)
      End If
   Next i
Next j
'Para la matriz sSFR_Operandos
For j = 0 To iMatriz
```

```
    For i = 0 To 19
        If sInstruccionesOperandos(j, 2) = sSFR_Operandos(i, 0) Then
            sInstruccionesOperandos(j, 2) = sSFR_Operandos(i, 1)
        End If
        If sInstruccionesOperandos(j, 3) = sSFR_Operandos(i, 0) Then
            sInstruccionesOperandos(j, 3) = sSFR_Operandos(i, 1)
        End If
    Next i
Next j
'Carga de las instrucciones a la matriz saMatrizInstrucciones
'y al ComboBox
FileNumber1 = FreeFile
Open sFileName1 For Input As #FileNumber1
iMatriz = -1
While Not EOF(FileNumber1)
    Line Input #FileNumber1, LineaLeida1
    iMatriz = iMatriz + 1
    saMatrizInstrucciones(iMatriz) = LineaLeida1
    cboInstrucciones.AddItem LineaLeida1 'Carga en el combobox cada linea leida del archivo
Wend
    cboInstrucciones.ListIndex = 0
Close #FileNumber1
'Código para la generacion de los que luego se cargaran a la ROM externa
ejec1 = -1
iROM = -1
While ejec1 < iMatriz
    ejec1 = ejec1 + 1
    PC = ejec1
    For ejec2 = 0 To 43
        If sInstruccionesOperandos(ejec1, 1) = sMatriz_Opcode(ejec2) Then
            Select Case ejec2
                Case 0
                    ' InstruccionDA
                    Y = 0
                    sInstruccionesOperandos(ejec1, 5) = "11010100"
                    sInstruccionesOperandos(ejec1, 8) = "1"
                    sInstruccionesOperandos(ejec1, 9) = "1"
                    Exit For
                Case 1
'                   InstruccionJC
                    Y = 0
                    sInstruccionesOperandos(ejec1, 5) = "01000000"
                    sInstruccionesOperandos(ejec1, 8) = "2"
                    sInstruccionesOperandos(ejec1, 9) = "2"
                    For ijc = 0 To 100
                        If sInstruccionesOperandos(ejec1, 2) = sSaltos_Rel(ijc) Then
                            iX1d_b = iSaltos_Rel(ijc)
                            Exit For
                        End If
                    Next ijc
                    ConversionD_B
                    For ijc1 = 0 To 7
                        sInstruccionesOperandos(ejec1, 6) = sInstruccionesOperandos(ejec1, 6) & iMatrizB(ijc1)
                    Next ijc1
                    'jc
                    Exit For
                Case 2
'                   InstruccionJB
                    Y = 0
                    iOperando = 2
                    Ubicacion_Bit
                    sInstruccionesOperandos(ejec1, 8) = "3"
                    sInstruccionesOperandos(ejec1, 9) = "2"
                    sInstruccionesOperandos(ejec1, 5) = "00100000"
                    If iNum_Byte >= 32 And iNum_Byte <= 47 Then
                        iX1d_b = (iNum_Byte - 32) * 8 - iNum_Bit + 7
                    Else
                        iX1d_b = iNum_Byte - iNum_Bit + 7
                    End If
                    ConversionD_B
                    For ijb = 0 To 7
```

```
            sInstruccionesOperandos(ejec1, 6) = sInstruccionesOperandos(ejec1, 6) & iMatrizB(ijb)
        Next ijb
        For ijb = 0 To 100
            If sInstruccionesOperandos(ejec1, 3) = sSaltos_Rel(ijb) Then
                iX1d_b = iSaltos_Rel(ijb)
                Exit For
            End If
        Next ijb
        ConversionD_B
        For ijb = 0 To 7
            sInstruccionesOperandos(ejec1, 7) = sInstruccionesOperandos(ejec1, 7) & iMatrizB(ijb)
        Next ijb
        'jb
        Exit For
    Case 3
        InstruccionJZ
        Y = 0
        sInstruccionesOperandos(ejec1, 8) = "2"
        sInstruccionesOperandos(ejec1, 9) = "2"
        sInstruccionesOperandos(ejec1, 5) = "01100000"
        For ijz = 0 To 100
            If sInstruccionesOperandos(ejec1, 2) = sSaltos_Rel(ijz) Then
                iX1d_b = iSaltos_Rel(ijz)
                Exit For
            End If
        Next ijz
        ConversionD_B
        For ijz = 0 To 7
            sInstruccionesOperandos(ejec1, 6) = sInstruccionesOperandos(ejec1, 6) & iMatrizB(ijz)
        Next ijz
        'jz
        Exit For
    Case 4
        InstruccionRL
        Y = 0
        sInstruccionesOperandos(ejec1, 8) = "1"
        sInstruccionesOperandos(ejec1, 9) = "1"
        sInstruccionesOperandos(ejec1, 5) = "00100011"
        'rl
        Exit For
    Case 5
        InstruccionRR
        Y = 0
        sInstruccionesOperandos(ejec1, 8) = "1"
        sInstruccionesOperandos(ejec1, 9) = "1"
        sInstruccionesOperandos(ejec1, 5) = "00000011"
        'rr
        Exit For
    Case 6
        InstruccionADD
        Y = 0
        kmov = 0
        UbiBancoRnData
        If sInstruccionesOperandos(ejec1, 2) = "A" Then
            For iadd = 1 To 10
                If sInstruccionesOperandos(ejec1, 3) = sMatriz_Operando12(iadd) Then
                    kmov = 1
                    sInstruccionesOperandos(ejec1, 8) = "1"
                    sInstruccionesOperandos(ejec1, 9) = "1"
                    Select Case iadd
                        Case 1 To 8
                            iX1d_b = iadd - 1
                            ConversionD_B
                            sInstruccionesOperandos(ejec1, 5) = "00101" & iMatrizB(5) & iMatrizB(6) & iMatrizB(7)
                            Exit For
                        Case 9 To 10
                            sInstruccionesOperandos(ejec1, 5) = "0010011" & (iadd - 9)
                            Exit For
                    End Select
                End If
            Next iadd
```

```
        If kmov = 0 Then
          sInstruccionesOperandos(ejec1, 8) = "2"
          sInstruccionesOperandos(ejec1, 9) = "1"
          LineaLeida1 = sInstruccionesOperandos(ejec1, 3)
          iLongitud1 = Len(LineaLeida1)
          Y = Y + 1
          LetrasLeidas = Mid(LineaLeida1, Y, 1)
          LetrasLeidas = UCase(LetrasLeidas)
          If LetrasLeidas = "#" Then
            sInstruccionesOperandos(ejec1, 5) = "00100100"
            Num_Data
            iX1d_b = iNum_Data
            ConversionD_B
            For Add = 0 To 7
              sInstruccionesOperandos(ejec1, 6) = sInstruccionesOperandos(ejec1, 6) & iMatrizB(Add)
            Next Add
          Else
            Y = Y - 1
            sInstruccionesOperandos(ejec1, 5) = "00100101"
            Num_Data
            iX1d_b = iNum_Data
            ConversionD_B
            For Add = 0 To 7
              sInstruccionesOperandos(ejec1, 6) = sInstruccionesOperandos(ejec1, 6) & iMatrizB(Add)
            Next Add
          End If
        End If
      End If
      'add
      Exit For
    Case 7
      InstruccionANL
      Y = 0
      kmov = 0
      kmov1 = 0
      kbit = 0
      UbiBancoRnData
      If sInstruccionesOperandos(ejec1, 2) = "A" Then
        kbit = 1
        kmov1 = 1
        For iadd = 1 To 10
          If sInstruccionesOperandos(ejec1, 3) = sMatriz_Operando12(iadd) Then
            sInstruccionesOperandos(ejec1, 8) = "1"
            sInstruccionesOperandos(ejec1, 9) = "1"
            kmov = 1
            Select Case iadd
              Case 1 To 8
                iX1d_b = iadd - 1
                ConversionD_B
                sInstruccionesOperandos(ejec1, 5) = "01011" & iMatrizB(5) & iMatrizB(6) & iMatrizB(7)
                Exit For
              Case 9 To 10
                sInstruccionesOperandos(ejec1, 5) = "0101011" & (iadd - 9)
                Exit For
            End Select
          End If
        Next iadd
        If kmov = 0 Then
          sInstruccionesOperandos(ejec1, 8) = "2"
          sInstruccionesOperandos(ejec1, 9) = "1"
          LineaLeida1 = sInstruccionesOperandos(ejec1, 3)
          iLongitud1 = Len(LineaLeida1)
          Y = Y + 1
          LetrasLeidas = Mid(LineaLeida1, Y, 1)
          LetrasLeidas = UCase(LetrasLeidas)
          If LetrasLeidas = "#" Then
            sInstruccionesOperandos(ejec1, 5) = "01010100"
            Num_Data
            iX1d_b = iNum_Data
            ConversionD_B
            For Add = 0 To 7
```

```
            sInstruccionesOperandos(ejec1, 6) = sInstruccionesOperandos(ejec1, 6) & iMatrizB(Add)
          Next Add
        Else
          Y = Y - 1
          sInstruccionesOperandos(ejec1, 5) = "01010101"
          Num_Data
          iX1d_b = iNum_Data
          ConversionD_B
          For Add = 0 To 7
            sInstruccionesOperandos(ejec1, 6) = sInstruccionesOperandos(ejec1, 6) & iMatrizB(Add)
          Next Add
        End If
      End If
    End If
    'Rutina para trabarjar a nivel de bits
    If kbit = 0 Then
      If sInstruccionesOperandos(ejec1, 2) = "C" Then
        kmov1 = 1
        sInstruccionesOperandos(ejec1, 8) = "2"
        sInstruccionesOperandos(ejec1, 9) = "2"
        LineaLeida1 = sInstruccionesOperandos(ejec1, 3)
        iLongitud1 = Len(LineaLeida1)
        LetrasLeidas = Mid(LineaLeida1, 1, 1)
        If LetrasLeidas = "/" Then
          sInstruccionesOperandos(ejec1, 5) = "10110000"
          LetrasLeidas = Mid(LineaLeida1, 2, iLongitud1)
          sInstruccionesOperandos(ejec1, 3) = LetrasLeidas
          iOperando = 3
          Ubicacion_Bit
          If iNum_Byte >= 32 And iNum_Byte <= 47 Then
            iX1d_b = (iNum_Byte - 32) * 8 - iNum_Bit + 7
          Else
            iX1d_b = iNum_Byte - iNum_Bit + 7
          End If
          ConversionD_B
          For Add = 0 To 7
            sInstruccionesOperandos(ejec1, 6) = sInstruccionesOperandos(ejec1, 6) & iMatrizB(Add)
          Next Add
          iMemoriaRAMint(208, 0) = iMemoriaRAMint(208, 0) And (Not (iMemoriaRAMint(iNum_Byte,
                                    iNum_Bit)))
          sInstruccionesOperandos(ejec1, 3) = "/" & sInstruccionesOperandos(ejec1, 3)
        Else
          sInstruccionesOperandos(ejec1, 5) = "10000010"
          iOperando = 3
          Ubicacion_Bit
          If iNum_Byte >= 32 And iNum_Byte <= 47 Then
            iX1d_b = (iNum_Byte - 32) * 8 - iNum_Bit + 7
          Else
            iX1d_b = iNum_Byte - iNum_Bit + 7
          End If
          ConversionD_B
          For Add = 0 To 7
            sInstruccionesOperandos(ejec1, 6) = sInstruccionesOperandos(ejec1, 6) & iMatrizB(Add)
          Next Add
        End If
      End If
    End If
    'Fin de la rutina de bits
    If kmov1 = 0 Then
      LineaLeida1 = sInstruccionesOperandos(ejec1, 2)
      iLongitud1 = Len(LineaLeida1)
      Num_Data
      iNum_Data1 = iNum_Data
      If sInstruccionesOperandos(ejec1, 3) = "A" Then
        kmov = 1
        sInstruccionesOperandos(ejec1, 8) = "2"
        sInstruccionesOperandos(ejec1, 9) = "1"
        sInstruccionesOperandos(ejec1, 5) = "01010010"
        iX1d_b = iNum_Data1
        ConversionD_B
        For Add = 0 To 7
```

```
            sInstruccionesOperandos(ejec1, 6) = sInstruccionesOperandos(ejec1, 6) & iMatrizB(Add)
          Next Add
        End If
        If kmov = 0 Then
          sInstruccionesOperandos(ejec1, 8) = "3"
          sInstruccionesOperandos(ejec1, 9) = "2"
          sInstruccionesOperandos(ejec1, 5) = "01010011"
          iX1d_b = iNum_Data1
          ConversionD_B
          For Add = 0 To 7
            sInstruccionesOperandos(ejec1, 6) = sInstruccionesOperandos(ejec1, 6) & iMatrizB(Add)
          Next Add
          LineaLeida1 = sInstruccionesOperandos(ejec1, 3)
          iLongitud1 = Len(LineaLeida1)
          Y = 0
          Y = Y + 1
          LetrasLeidas = Mid(LineaLeida1, Y, 1)
          LetrasLeidas = UCase(LetrasLeidas)
          If LetrasLeidas = "#" Then
            Num_Data
            iX1d_b = iNum_Data
            ConversionD_B
            For Add = 0 To 7
              sInstruccionesOperandos(ejec1, 7) = sInstruccionesOperandos(ejec1, 7) & iMatrizB(Add)
            Next Add
          End If
        End If
      End If
      'anl
      Exit For
    Case 8
      InstruccionCLR
      Y = 0
      If sInstruccionesOperandos(ejec1, 2) = "A" Then
        sInstruccionesOperandos(ejec1, 8) = "1"
        sInstruccionesOperandos(ejec1, 9) = "1"
        sInstruccionesOperandos(ejec1, 5) = "11100100"
      ElseIf sInstruccionesOperandos(ejec1, 2) = "C" Then
        sInstruccionesOperandos(ejec1, 8) = "1"
        sInstruccionesOperandos(ejec1, 9) = "1"
        sInstruccionesOperandos(ejec1, 5) = "11000011"
      Else
        sInstruccionesOperandos(ejec1, 8) = "2"
        sInstruccionesOperandos(ejec1, 9) = "1"
        sInstruccionesOperandos(ejec1, 5) = "11000010"
        iOperando = 2
        Ubicacion_Bit
        If iNum_Byte >= 32 And iNum_Byte <= 47 Then
          iX1d_b = (iNum_Byte - 32) * 8 - iNum_Bit + 7
        Else
          iX1d_b = iNum_Byte - iNum_Bit + 7
        End If
        ConversionD_B
        For Add = 0 To 7
          sInstruccionesOperandos(ejec1, 6) = sInstruccionesOperandos(ejec1, 6) & iMatrizB(Add)
        Next Add
      End If
      'clr
      Exit For
    Case 9
      InstruccionCPL
      Y = 0
      If sInstruccionesOperandos(ejec1, 2) = "A" Then
        sInstruccionesOperandos(ejec1, 8) = "1"
        sInstruccionesOperandos(ejec1, 9) = "1"
        sInstruccionesOperandos(ejec1, 5) = "11110100"
      ElseIf sInstruccionesOperandos(ejec1, 2) = "C" Then
        sInstruccionesOperandos(ejec1, 8) = "1"
        sInstruccionesOperandos(ejec1, 9) = "1"
        sInstruccionesOperandos(ejec1, 5) = "10110011"
      Else
```

```
            sInstruccionesOperandos(ejec1, 8) = "2"
            sInstruccionesOperandos(ejec1, 9) = "1"
            sInstruccionesOperandos(ejec1, 5) = "10110010"
            iOperando = 2
            Ubicacion_Bit
            If iNum_Byte >= 32 And iNum_Byte <= 47 Then
               iX1d_b = (iNum_Byte - 32) * 8 - iNum_Bit + 7
            Else
               iX1d_b = iNum_Byte - iNum_Bit + 7
            End If
            ConversionD_B
            For Add = 0 To 7
               sInstruccionesOperandos(ejec1, 6) = sInstruccionesOperandos(ejec1, 6) & iMatrizB(Add)
            Next Add
         End If
         'cpl
         Exit For
      Case 10
         InstruccionDIV
         Y = 0
         sInstruccionesOperandos(ejec1, 8) = "1"
         sInstruccionesOperandos(ejec1, 9) = "4"
         sInstruccionesOperandos(ejec1, 5) = "10000100"
         'div
         Exit For
      Case 11
         InstruccionDEC
         Y = 0
         kmov = 0
         UbiBancoRnData
         sInstruccionesOperandos(ejec1, 9) = "1"
         For iadd = 0 To 10
            If sInstruccionesOperandos(ejec1, 2) = sMatriz_Operando12(iadd) Then
               kmov = 1
               Select Case iadd
                  Case 0
                     sInstruccionesOperandos(ejec1, 8) = "1"
                     sInstruccionesOperandos(ejec1, 5) = "00010100"
                     Exit For
                  Case 1 To 8
                     sInstruccionesOperandos(ejec1, 8) = "1"
                     iX1d_b = iadd - 1
                     ConversionD_B
                     sInstruccionesOperandos(ejec1, 5) = "00011" & iMatrizB(5) & iMatrizB(6) & iMatrizB(7)
                  Case 9 To 10
                     sInstruccionesOperandos(ejec1, 8) = "1"
                     sInstruccionesOperandos(ejec1, 5) = "0001011" & (iadd - 9)
                     Exit For
               End Select
            End If
         Next iadd
         If kmov = 0 Then
            sInstruccionesOperandos(ejec1, 8) = "2"
            sInstruccionesOperandos(ejec1, 5) = "00010101"
            LineaLeida1 = sInstruccionesOperandos(ejec1, 2)
            iLongitud1 = Len(LineaLeida1)
            Num_Data
            iX1d_b = iNum_Data
            ConversionD_B
            For Add = 0 To 7
               sInstruccionesOperandos(ejec1, 6) = sInstruccionesOperandos(ejec1, 6) & iMatrizB(Add)
            Next Add
         End If
         'dec
         Exit For
      Case 12
         InstruccionINC
         Y = 0
         kmov = 0
         UbiBancoRnData
         sInstruccionesOperandos(ejec1, 9) = "1"
```

```
For iadd = 0 To 11
  If sInstruccionesOperandos(ejec1, 2) = sMatriz_Operando12(iadd) Then
    kmov = 1
    Select Case iadd
      Case 0
        sInstruccionesOperandos(ejec1, 8) = "1"
        sInstruccionesOperandos(ejec1, 9) = "1"
        sInstruccionesOperandos(ejec1, 5) = "00000100"
        Exit For
      Case 1 To 8
        sInstruccionesOperandos(ejec1, 8) = "1"
        sInstruccionesOperandos(ejec1, 9) = "1"
        iX1d_b = iadd - 1
        ConversionD_B
        sInstruccionesOperandos(ejec1, 5) = "00001" & iMatrizB(5) & iMatrizB(6) & iMatrizB(7)
      Case 9 To 10
        sInstruccionesOperandos(ejec1, 8) = "1"
        sInstruccionesOperandos(ejec1, 9) = "1"
        sInstruccionesOperandos(ejec1, 5) = "0000011" & (iadd - 9)
        Exit For
      Case 11
        sInstruccionesOperandos(ejec1, 8) = "1"
        sInstruccionesOperandos(ejec1, 9) = "2"
        sInstruccionesOperandos(ejec1, 5) = "10100011"
        Exit For
    End Select
  End If
Next iadd
If kmov = 0 Then
  sInstruccionesOperandos(ejec1, 8) = "2"
  sInstruccionesOperandos(ejec1, 9) = "1"
  sInstruccionesOperandos(ejec1, 5) = "00000101"
  LineaLeida1 = sInstruccionesOperandos(ejec1, 2)
  iLongitud1 = Len(LineaLeida1)
  Num_Data
  iX1d_b = iNum_Data
  ConversionD_B
  For Add = 0 To 7
    sInstruccionesOperandos(ejec1, 6) = sInstruccionesOperandos(ejec1, 6) & iMatrizB(Add)
  Next Add
End If
'inc
Exit For
Case 13
  InstruccionJNC
  Y = 0
  sInstruccionesOperandos(ejec1, 8) = "2"
  sInstruccionesOperandos(ejec1, 9) = "2"
  sInstruccionesOperandos(ejec1, 5) = "01010000"
  For ijnc = 0 To 29
    If sInstruccionesOperandos(ejec1, 2) = sSaltos_Rel(ijnc) Then
      ijnc1 = iSaltos_Rel(ijnc) - 1
      Exit For
    End If
  Next ijnc
  iX1d_b = ijnc1
  ConversionD_B
  For Add = 0 To 7
    sInstruccionesOperandos(ejec1, 6) = sInstruccionesOperandos(ejec1, 6) & iMatrizB(Add)
  Next Add
End If
'jnc
Exit For
Case 14
  InstruccionJNB
  Y = 0
  sInstruccionesOperandos(ejec1, 8) = "3"
  sInstruccionesOperandos(ejec1, 9) = "2"
  sInstruccionesOperandos(ejec1, 5) = "00110000"
  iOperando = 2
  Ubicacion_Bit
```

```
        If iNum_Byte >= 32 And iNum_Byte <= 47 Then
          iX1d_b = (iNum_Byte - 32) * 8 - iNum_Bit + 7
        Else
          iX1d_b = iNum_Byte - iNum_Bit + 7
        End If
        ConversionD_B
        For Add = 0 To 7
          sInstruccionesOperandos(ejec1, 6) = sInstruccionesOperandos(ejec1, 6) & iMatrizB(Add)
        Next Add
        For ijnb = 0 To 29
          If sInstruccionesOperandos(ejec1, 3) = sSaltos_Rel(ijnb) Then
            ijnb1 = iSaltos_Rel(ijnb) - 1
            Exit For
          End If
        Next ijnb
        iX1d_b = ijnb1
        ConversionD_B
        For Add = 0 To 7
          sInstruccionesOperandos(ejec1, 7) = sInstruccionesOperandos(ejec1, 7) & iMatrizB(Add)
        Next Add
        'jnb
        Exit For
      Case 15
'       InstruccionJBC
        Y = 0
        sInstruccionesOperandos(ejec1, 8) = "3"
        sInstruccionesOperandos(ejec1, 9) = "2"
        sInstruccionesOperandos(ejec1, 5) = "00010000"
        iOperando = 2
        Ubicacion_Bit
        If iNum_Byte >= 32 And iNum_Byte <= 47 Then
          iX1d_b = (iNum_Byte - 32) * 8 - iNum_Bit + 7
        Else
          iX1d_b = iNum_Byte - iNum_Bit + 7
        End If
        ConversionD_B
        For Add = 0 To 7
          sInstruccionesOperandos(ejec1, 6) = sInstruccionesOperandos(ejec1, 6) & iMatrizB(Add)
        Next Add
        For ijnb = 0 To 29
          If sInstruccionesOperandos(ejec1, 3) = sSaltos_Rel(ijnb) Then
            ijnb1 = iSaltos_Rel(ijnb) - 1
            Exit For
          End If
        Next ijnb
        iX1d_b = ijnb1
        ConversionD_B
        For Add = 0 To 7
          sInstruccionesOperandos(ejec1, 7) = sInstruccionesOperandos(ejec1, 7) & iMatrizB(Add)
        Next Add
        'jbc
        Exit For
      Case 16
'       InstruccionJMP
        Y = 0
        sInstruccionesOperandos(ejec1, 8) = "1"
        sInstruccionesOperandos(ejec1, 9) = "2"
        sInstruccionesOperandos(ejec1, 5) = "01110011"
        'jmp
        Exit For
      Case 17
'       InstruccionJNZ
        Y = 0
        sInstruccionesOperandos(ejec1, 8) = "2"
        sInstruccionesOperandos(ejec1, 9) = "2"
        sInstruccionesOperandos(ejec1, 5) = "01110000"
        For ijnz = 0 To 29
          If sInstruccionesOperandos(ejec1, 2) = sSaltos_Rel(ijnz) Then
            ijnz1 = iSaltos_Rel(ijnz) - 1
            Exit For
          End If
```

```
            Next ijnz
            iX1d_b = ijnz1
            ConversionD_B
            For Add = 0 To 7
              sInstruccionesOperandos(ejec1, 6) = sInstruccionesOperandos(ejec1, 6) & iMatrizB(Add)
            Next Add
            'jnz
            Exit For
          Case 18
'           InstruccionMUL
            Y = 0
            sInstruccionesOperandos(ejec1, 8) = "1"
            sInstruccionesOperandos(ejec1, 9) = "4"
            sInstruccionesOperandos(ejec1, 5) = "10100100"
            'mul
            Exit For
          Case 19
'           InstruccionMOV
            Y = 0
            kmov = 0
            kmov1 = 0
            kbit = 0
            UbiBancoRnData
            For iadd1 = 0 To 11
              If sInstruccionesOperandos(ejec1, 2) = sMatriz_Operando12(iadd1) Then
              kbit = 1
              kmov1 = 1
              Select Case iadd1
                Case 0
                  For iadd = 1 To 10
                    If sInstruccionesOperandos(ejec1, 3) = sMatriz_Operando12(iadd) Then
                      kmov = 1
                      sInstruccionesOperandos(ejec1, 8) = "1"
                      sInstruccionesOperandos(ejec1, 9) = "1"
                      Select Case iadd
                        Case 1 To 8
                          iX1d_b = iadd - 1
                          ConversionD_B
                          sInstruccionesOperandos(ejec1, 5) = "11111" & iMatrizB(5) & iMatrizB(6) & iMatrizB(7)
                          Exit For   'iadd
                        Case 9 To 10
                          sInstruccionesOperandos(ejec1, 5) = "1110011" & (iadd - 9)
                          Exit For   'iadd
                      End Select
                    End If
                  Next iadd
                  If kmov = 0 Then
                    sInstruccionesOperandos(ejec1, 8) = "2"
                    sInstruccionesOperandos(ejec1, 9) = "1"
                    LineaLeida1 = sInstruccionesOperandos(ejec1, 3)
                    iLongitud1 = Len(LineaLeida1)
                    Y = Y + 1
                    LetrasLeidas = Mid(LineaLeida1, Y, 1)
                    LetrasLeidas = UCase(LetrasLeidas)
                    If LetrasLeidas = "#" Then
                      sInstruccionesOperandos(ejec1, 5) = "01110100"
                      Num_Data
                      iX1d_b = iNum_Data
                      ConversionD_B
                      For Add = 0 To 7
                        sInstruccionesOperandos(ejec1, 6) = sInstruccionesOperandos(ejec1, 6) & iMatrizB(Add)
                      Next Add
                    Else
                      Y = Y - 1
                      sInstruccionesOperandos(ejec1, 5) = "11100101"
                      Num_Data
                      iX1d_b = iNum_Data
                      ConversionD_B
                      For Add = 0 To 7
                        sInstruccionesOperandos(ejec1, 6) = sInstruccionesOperandos(ejec1, 6) & iMatrizB(Add)
                      Next Add
```

```
        End If
      End If
      Exit For   'iadd1
    Case 1 To 8
      If sInstruccionesOperandos(ejec1, 3) = "A" Then
        kmov = 1
        sInstruccionesOperandos(ejec1, 8) = "1"
        sInstruccionesOperandos(ejec1, 9) = "1"
        iX1d_b = iadd1 - 1
        ConversionD_B
        sInstruccionesOperandos(ejec1, 5) = "11111" & iMatrizB(5) & iMatrizB(6) & iMatrizB(7)
      End If
      If kmov = 0 Then
        sInstruccionesOperandos(ejec1, 8) = "2"
        LineaLeida1 = sInstruccionesOperandos(ejec1, 3)
        iLongitud1 = Len(LineaLeida1)
        Y = Y + 1
        LetrasLeidas = Mid(LineaLeida1, Y, 1)
        LetrasLeidas = UCase(LetrasLeidas)
        If LetrasLeidas = "#" Then
          sInstruccionesOperandos(ejec1, 9) = "1"
          iX1d_b = iadd1 - 1
          ConversionD_B
          sInstruccionesOperandos(ejec1, 5) = "01111" & iMatrizB(5) & iMatrizB(6) & iMatrizB(7)
          Num_Data
          iX1d_b = iNum_Data
          ConversionD_B
          For Add = 0 To 7
            sInstruccionesOperandos(ejec1, 6) = sInstruccionesOperandos(ejec1, 6) & iMatrizB(Add)
          Next Add
        Else
          Y = Y - 1
          sInstruccionesOperandos(ejec1, 9) = "2"
          iX1d_b = iadd1 - 1
          ConversionD_B
          sInstruccionesOperandos(ejec1, 5) = "10101" & iMatrizB(5) & iMatrizB(6) & iMatrizB(7)
          Num_Data
          iX1d_b = iNum_Data
          ConversionD_B
          For Add = 0 To 7
            sInstruccionesOperandos(ejec1, 6) = sInstruccionesOperandos(ejec1, 6) & iMatrizB(Add)
          Next Add
        End If
      End If
      Exit For   'iadd1
    Case 9 To 10
      If sInstruccionesOperandos(ejec1, 3) = "A" Then
        kmov = 1
        sInstruccionesOperandos(ejec1, 8) = "1"
        sInstruccionesOperandos(ejec1, 9) = "1"
        sInstruccionesOperandos(ejec1, 5) = "1110011" & (iadd1 - 9)
      End If
      If kmov = 0 Then
        sInstruccionesOperandos(ejec1, 8) = "2"
        LineaLeida1 = sInstruccionesOperandos(ejec1, 3)
        iLongitud1 = Len(LineaLeida1)
        Y = Y + 1
        LetrasLeidas = Mid(LineaLeida1, Y, 1)
        LetrasLeidas = UCase(LetrasLeidas)
        If LetrasLeidas = "#" Then
          sInstruccionesOperandos(ejec1, 9) = "1"
          sInstruccionesOperandos(ejec1, 5) = "0111011" & (iadd1 - 9)
          Num_Data
          iX1d_b = iNum_Data
          ConversionD_B
          For Add = 0 To 7
            sInstruccionesOperandos(ejec1, 6) = sInstruccionesOperandos(ejec1, 6) & iMatrizB(Add)
          Next Add
        Else
          Y = Y - 1
          sInstruccionesOperandos(ejec1, 9) = "2"
```

```
                sInstruccionesOperandos(ejec1, 5) = "1010011" & (iadd1 - 9)
                Num_Data
                iX1d_b = iNum_Data
                ConversionD_B
                For Add = 0 To 7
                  sInstruccionesOperandos(ejec1, 6) = sInstruccionesOperandos(ejec1, 6) & iMatrizB(Add)
                Next Add
              End If
            End If
            Exit For   'iadd1
          Case 11
            LineaLeida1 = sInstruccionesOperandos(ejec1, 3)
            sInstruccionesOperandos(ejec1, 8) = "3"
            sInstruccionesOperandos(ejec1, 9) = "2"
            sInstruccionesOperandos(ejec1, 5) = "10010000"
            Num_Data16
            i16Bits1 = iNum_Data16
            ConversionDPTRD_B
            For Add = 0 To 7
              sInstruccionesOperandos(ejec1, 6) = sInstruccionesOperandos(ejec1, 6) & iBitDPTR(Add)
              sInstruccionesOperandos(ejec1, 7) = sInstruccionesOperandos(ejec1, 7) & iBitDPTR(Add + 8)
            Next Add
            Exit For   'iadd1
        End Select
        End If
      Next iadd1
      'Rutina para manejo de Bits
      If kbit = 0 Then
        If sInstruccionesOperandos(ejec1, 2) = "C" Or sInstruccionesOperandos(ejec1, 3) = "C" Then
          kmov1 = 1
          If sInstruccionesOperandos(ejec1, 2) = "C" Then
            sInstruccionesOperandos(ejec1, 8) = "2"
            sInstruccionesOperandos(ejec1, 9) = "1"
            iOperando = 3
            Ubicacion_Bit
            sInstruccionesOperandos(ejec1, 5) = "10100010"
            If iNum_Byte >= 32 And iNum_Byte <= 47 Then
              iX1d_b = (iNum_Byte - 32) * 8 - iNum_Bit + 7
            Else
              iX1d_b = iNum_Byte - iNum_Bit + 7
            End If
            ConversionD_B
            For Add = 0 To 7
              sInstruccionesOperandos(ejec1, 6) = sInstruccionesOperandos(ejec1, 6) & iMatrizB(Add)
            Next Add
          Else
            sInstruccionesOperandos(ejec1, 8) = "2"
            sInstruccionesOperandos(ejec1, 9) = "2"
            iOperando = 2
            Ubicacion_Bit
            sInstruccionesOperandos(ejec1, 5) = "10010010"
            If iNum_Byte >= 32 And iNum_Byte <= 47 Then
              iX1d_b = (iNum_Byte - 32) * 8 - iNum_Bit + 7
            Else
              iX1d_b = iNum_Byte - iNum_Bit + 7
            End If
            ConversionD_B
            For Add = 0 To 7
              sInstruccionesOperandos(ejec1, 6) = sInstruccionesOperandos(ejec1, 6) & iMatrizB(Add)
            Next Add
          End If
        End If
      End If
      'Fin de rutina para bits
      If kmov1 = 0 Then
        LineaLeida1 = sInstruccionesOperandos(ejec1, 2)
        iLongitud1 = Len(LineaLeida1)
        Num_Data
        iNum_Data1 = iNum_Data
        For iadd = 0 To 10
          If sInstruccionesOperandos(ejec1, 3) = sMatriz_Operando12(iadd) Then
```

```
        kmov = 1
        Select Case iadd
          Case 0
            sInstruccionesOperandos(ejec1, 8) = "2"
            sInstruccionesOperandos(ejec1, 9) = "1"
            sInstruccionesOperandos(ejec1, 5) = "11110101"
            iX1d_b = iNum_Data1
            ConversionD_B
            For Add = 0 To 7
              sInstruccionesOperandos(ejec1, 6) = sInstruccionesOperandos(ejec1, 6) & iMatrizB(Add)
            Next Add
            Exit For   'iadd
          Case 1 To 8
            sInstruccionesOperandos(ejec1, 8) = "2"
            sInstruccionesOperandos(ejec1, 9) = "2"
            iX1d_b = iNum_Data1
            ConversionD_B
            For Add = 0 To 7
              sInstruccionesOperandos(ejec1, 6) = sInstruccionesOperandos(ejec1, 6) & iMatrizB(Add)
            Next Add
            iX1d_b = iadd - 1
            ConversionD_B
            sInstruccionesOperandos(ejec1, 5) = "10001" & iMatrizB(5) & iMatrizB(6) & iMatrizB(7)
            Exit For   'iadd
          Case 9 To 10
            sInstruccionesOperandos(ejec1, 8) = "2"
            sInstruccionesOperandos(ejec1, 9) = "2"
            iX1d_b = iNum_Data1
            ConversionD_B
            For Add = 0 To 7
              sInstruccionesOperandos(ejec1, 6) = sInstruccionesOperandos(ejec1, 6) & iMatrizB(Add)
            Next Add
            sInstruccionesOperandos(ejec1, 5) = "1000011" & (iadd1 - 9)
            Exit For   'iadd
        End Select
      End If
    Next iadd
    If kmov = 0 Then
      Y = 0
      LineaLeida1 = sInstruccionesOperandos(ejec1, 3)
      iLongitud1 = Len(LineaLeida1)
      Y = Y + 1
      LetrasLeidas = Mid(LineaLeida1, Y, 1)
      LetrasLeidas = UCase(LetrasLeidas)
      If LetrasLeidas = "#" Then
        sInstruccionesOperandos(ejec1, 8) = "3"
        sInstruccionesOperandos(ejec1, 9) = "2"
        sInstruccionesOperandos(ejec1, 5) = "01110101"
        iX1d_b = iNum_Data1
        ConversionD_B
        For Add = 0 To 7
          sInstruccionesOperandos(ejec1, 6) = sInstruccionesOperandos(ejec1, 6) & iMatrizB(Add)
        Next Add
        Num_Data
        iX1d_b = iNum_Data
        ConversionD_B
        For Add = 0 To 7
          sInstruccionesOperandos(ejec1, 7) = sInstruccionesOperandos(ejec1, 7) & iMatrizB(Add)
        Next Add
      Else
        sInstruccionesOperandos(ejec1, 8) = "3"
        sInstruccionesOperandos(ejec1, 9) = "2"
        sInstruccionesOperandos(ejec1, 5) = "10000101"
        iX1d_b = iNum_Data1
        ConversionD_B
        For Add = 0 To 7
          sInstruccionesOperandos(ejec1, 7) = sInstruccionesOperandos(ejec1, 7) & iMatrizB(Add)
        Next Add
        Y = Y - 1
        Num_Data
        iX1d_b = iNum_Data
```

```
                ConversionD_B
                For Add = 0 To 7
                  sInstruccionesOperandos(ejec1, 6) = sInstruccionesOperandos(ejec1, 6) & iMatrizB(Add)
                Next Add
              End If
            End If
          End If
          'mov
          Exit For
        Case 20
          InstruccionNOP
          Y = 0
          sInstruccionesOperandos(ejec1, 8) = "1"
          sInstruccionesOperandos(ejec1, 9) = "1"
          sInstruccionesOperandos(ejec1, 5) = "00000000"
          'nop
          Exit For
        Case 21
          InstruccionORL
          Y = 0
          kmov = 0
          kmov1 = 0
          kbit = 0
          UbiBancoRnData
          If sInstruccionesOperandos(ejec1, 2) = "A" Then
            kbit = 1
            kmov1 = 1
            For iadd = 1 To 10
              If sInstruccionesOperandos(ejec1, 3) = sMatriz_Operando12(iadd) Then
                sInstruccionesOperandos(ejec1, 8) = "1"
                sInstruccionesOperandos(ejec1, 9) = "1"
                kmov = 1
                Select Case iadd
                  Case 1 To 8
                    iX1d_b = iadd - 1
                    ConversionD_B
                    sInstruccionesOperandos(ejec1, 5) = "01001" & iMatrizB(5) & iMatrizB(6) & iMatrizB(7)
                    Exit For
                  Case 9 To 10
                    sInstruccionesOperandos(ejec1, 5) = "0100011" & (iadd - 9)
                    Exit For
                End Select
              End If
            Next iadd
            If kmov = 0 Then
              sInstruccionesOperandos(ejec1, 8) = "2"
              sInstruccionesOperandos(ejec1, 9) = "1"
              LineaLeida1 = sInstruccionesOperandos(ejec1, 3)
              iLongitud1 = Len(LineaLeida1)
              Y = Y + 1
              LetrasLeidas = Mid(LineaLeida1, Y, 1)
              LetrasLeidas = UCase(LetrasLeidas)
              If LetrasLeidas = "#" Then
                sInstruccionesOperandos(ejec1, 5) = "01000100"
                Num_Data
                iX1d_b = iNum_Data
                ConversionD_B
                For Add = 0 To 7
                  sInstruccionesOperandos(ejec1, 6) = sInstruccionesOperandos(ejec1, 6) & iMatrizB(Add)
                Next Add
              Else
                Y = Y - 1
                sInstruccionesOperandos(ejec1, 5) = "01000101"
                Num_Data
                iX1d_b = iNum_Data
                ConversionD_B
                For Add = 0 To 7
                  sInstruccionesOperandos(ejec1, 6) = sInstruccionesOperandos(ejec1, 6) & iMatrizB(Add)
                Next Add
              End If
            End If
```

```
End If
'Rutina para trabarjar a nivel de bits
If kbit = 0 Then
  If sInstruccionesOperandos(ejec1, 2) = "C" Then
    kmov1 = 1
    sInstruccionesOperandos(ejec1, 8) = "2"
    sInstruccionesOperandos(ejec1, 9) = "2"
    LineaLeida1 = sInstruccionesOperandos(ejec1, 3)
    iLongitud1 = Len(LineaLeida1)
    LetrasLeidas = Mid(LineaLeida1, 1, 1)
    If LetrasLeidas = "/" Then
      sInstruccionesOperandos(ejec1, 5) = "10100000"
      LetrasLeidas = Mid(LineaLeida1, 2, iLongitud1)
      sInstruccionesOperandos(ejec1, 3) = LetrasLeidas
      iOperando = 3
      Ubicacion_Bit
      If iNum_Byte >= 32 And iNum_Byte <= 47 Then
        iX1d_b = (iNum_Byte - 32) * 8 - iNum_Bit + 7
      Else
        iX1d_b = iNum_Byte - iNum_Bit + 7
      End If
      ConversionD_B
      For Add = 0 To 7
        sInstruccionesOperandos(ejec1, 6) = sInstruccionesOperandos(ejec1, 6) & iMatrizB(Add)
      Next Add
      iMemoriaRAMint(208, 0) = iMemoriaRAMint(208, 0) And (Not (iMemoriaRAMint(iNum_Byte,
                                iNum_Bit)))
      sInstruccionesOperandos(ejec1, 3) = "/" & sInstruccionesOperandos(ejec1, 3)
    Else
      sInstruccionesOperandos(ejec1, 5) = "01110010"
      iOperando = 3
      Ubicacion_Bit
      If iNum_Byte >= 32 And iNum_Byte <= 47 Then
        iX1d_b = (iNum_Byte - 32) * 8 - iNum_Bit + 7
      Else
        iX1d_b = iNum_Byte - iNum_Bit + 7
      End If
      ConversionD_B
      For Add = 0 To 7
        sInstruccionesOperandos(ejec1, 6) = sInstruccionesOperandos(ejec1, 6) & iMatrizB(Add)
      Next Add
    End If
  End If
End If
'Fin de la rutina de bits
If kmov1 = 0 Then
  LineaLeida1 = sInstruccionesOperandos(ejec1, 2)
  iLongitud1 = Len(LineaLeida1)
  Num_Data
  iNum_Data1 = iNum_Data
  If sInstruccionesOperandos(ejec1, 3) = "A" Then
    kmov = 1
    sInstruccionesOperandos(ejec1, 8) = "2"
    sInstruccionesOperandos(ejec1, 9) = "1"
    sInstruccionesOperandos(ejec1, 5) = "01000010"
    iX1d_b = iNum_Data1
    ConversionD_B
    For Add = 0 To 7
      sInstruccionesOperandos(ejec1, 6) = sInstruccionesOperandos(ejec1, 6) & iMatrizB(Add)
    Next Add
  End If
  If kmov = 0 Then
    sInstruccionesOperandos(ejec1, 8) = "3"
    sInstruccionesOperandos(ejec1, 9) = "2"
    sInstruccionesOperandos(ejec1, 5) = "01000011"
    iX1d_b = iNum_Data1
    ConversionD_B
    For Add = 0 To 7
      sInstruccionesOperandos(ejec1, 6) = sInstruccionesOperandos(ejec1, 6) & iMatrizB(Add)
    Next Add
    LineaLeida1 = sInstruccionesOperandos(ejec1, 3)
```

```
            iLongitud1 = Len(LineaLeida1)
            Y = 0
            Y = Y + 1
            LetrasLeidas = Mid(LineaLeida1, Y, 1)
            LetrasLeidas = UCase(LetrasLeidas)
            If LetrasLeidas = "#" Then
              Num_Data
              iX1d_b = iNum_Data
              ConversionD_B
              For Add = 0 To 7
                sInstruccionesOperandos(ejec1, 7) = sInstruccionesOperandos(ejec1, 7) & iMatrizB(Add)
              Next Add
            End If
          End If
        End If
        'orl
        Exit For
      Case 22
        InstruccionPOP
        Y = 0
        sInstruccionesOperandos(ejec1, 8) = "2"
        sInstruccionesOperandos(ejec1, 9) = "2"
        sInstruccionesOperandos(ejec1, 5) = "11010000"
        LineaLeida1 = sInstruccionesOperandos(ejec1, 2)
        iLongitud1 = Len(LineaLeida1)
        Num_Data
        If sInstruccionesOperandos(ejec1, 2) = "A" Then
          iX1d_b = 224
        Else
          iX1d_b = iNum_Data
        End If
        ConversionD_B
        For Add = 0 To 7
          sInstruccionesOperandos(ejec1, 6) = sInstruccionesOperandos(ejec1, 6) & iMatrizB(Add)
        Next Add
        'pop
        Exit For
      Case 23
        InstruccionRLC
        Y = 0
        sInstruccionesOperandos(ejec1, 8) = "1"
        sInstruccionesOperandos(ejec1, 9) = "1"
        sInstruccionesOperandos(ejec1, 5) = "00110011"
        'rlc
        Exit For
      Case 24
        InstruccionRRC
        Y = 0
        sInstruccionesOperandos(ejec1, 8) = "1"
        sInstruccionesOperandos(ejec1, 9) = "1"
        sInstruccionesOperandos(ejec1, 5) = "00010011"
        'rrc
        Exit For
      Case 25
        InstruccionRET
        Y = 0
        sInstruccionesOperandos(ejec1, 8) = "1"
        sInstruccionesOperandos(ejec1, 9) = "2"
        sInstruccionesOperandos(ejec1, 5) = "00100010"
        'ret
        Exit For
      Case 26
        InstruccionXRL
        Y = 0
        kmov = 0
        kmov1 = 0
        UbiBancoRnData
        If sInstruccionesOperandos(ejec1, 2) = "A" Then
          kmov1 = 1
          For iadd = 1 To 10
            If sInstruccionesOperandos(ejec1, 3) = sMatriz_Operando12(iadd) Then
```

```
            sInstruccionesOperandos(ejec1, 8) = "1"
            sInstruccionesOperandos(ejec1, 9) = "1"
            kmov = 1
            Select Case iadd
              Case 1 To 8
                iX1d_b = iadd - 1
                ConversionD_B
                sInstruccionesOperandos(ejec1, 5) = "01101" & iMatrizB(5) & iMatrizB(6) & iMatrizB(7)
                Exit For
              Case 9 To 10
                sInstruccionesOperandos(ejec1, 5) = "0110011" & (iadd - 9)
                Exit For
            End Select
          End If
        Next iadd
        If kmov = 0 Then
          sInstruccionesOperandos(ejec1, 8) = "2"
          sInstruccionesOperandos(ejec1, 9) = "1"
          LineaLeida1 = sInstruccionesOperandos(ejec1, 3)
          iLongitud1 = Len(LineaLeida1)
          Y = Y + 1
          LetrasLeidas = Mid(LineaLeida1, Y, 1)
          LetrasLeidas = UCase(LetrasLeidas)
          If LetrasLeidas = "#" Then
            sInstruccionesOperandos(ejec1, 5) = "01100100"
            Num_Data
            iX1d_b = iNum_Data
            ConversionD_B
            For Add = 0 To 7
              sInstruccionesOperandos(ejec1, 6) = sInstruccionesOperandos(ejec1, 6) & iMatrizB(Add)
            Next Add
          Else
            Y = Y - 1
            sInstruccionesOperandos(ejec1, 5) = "01100101"
            Num_Data
            iX1d_b = iNum_Data
            ConversionD_B
            For Add = 0 To 7
              sInstruccionesOperandos(ejec1, 6) = sInstruccionesOperandos(ejec1, 6) & iMatrizB(Add)
            Next Add
          End If
        End If
      End If
      If kmov1 = 0 Then
        LineaLeida1 = sInstruccionesOperandos(ejec1, 2)
        iLongitud1 = Len(LineaLeida1)
        Num_Data
        iNum_Data1 = iNum_Data
        If sInstruccionesOperandos(ejec1, 3) = "A" Then
          kmov = 1
          sInstruccionesOperandos(ejec1, 8) = "2"
          sInstruccionesOperandos(ejec1, 9) = "1"
          sInstruccionesOperandos(ejec1, 5) = "01100010"
          iX1d_b = iNum_Data1
          ConversionD_B
          For Add = 0 To 7
            sInstruccionesOperandos(ejec1, 6) = sInstruccionesOperandos(ejec1, 6) & iMatrizB(Add)
          Next Add
        End If
        If kmov = 0 Then
          sInstruccionesOperandos(ejec1, 8) = "3"
          sInstruccionesOperandos(ejec1, 9) = "2"
          sInstruccionesOperandos(ejec1, 5) = "01100011"
          iX1d_b = iNum_Data1
          ConversionD_B
          For Add = 0 To 7
            sInstruccionesOperandos(ejec1, 6) = sInstruccionesOperandos(ejec1, 6) & iMatrizB(Add)
          Next Add
          LineaLeida1 = sInstruccionesOperandos(ejec1, 3)
          iLongitud1 = Len(LineaLeida1)
          Y = 0
```

```
            Y = Y + 1
            LetrasLeidas = Mid(LineaLeida1, Y, 1)
            LetrasLeidas = UCase(LetrasLeidas)
            If LetrasLeidas = "#" Then
              Num_Data
              iX1d_b = iNum_Data
              ConversionD_B
              For Add = 0 To 7
                sInstruccionesOperandos(ejec1, 7) = sInstruccionesOperandos(ejec1, 7) & iMatrizB(Add)
              Next Add
            End If
          End If
        End If
        'xrl
        Exit For
      Case 27
'       InstruccionXCH
        Y = 0
        kmov = 0
        UbiBancoRnData
        If sInstruccionesOperandos(ejec1, 2) = "A" Then
          For iadd = 1 To 10
            If sInstruccionesOperandos(ejec1, 3) = sMatriz_Operando12(iadd) Then
              kmov = 1
              sInstruccionesOperandos(ejec1, 8) = "1"
              sInstruccionesOperandos(ejec1, 9) = "1"
              Select Case iadd
                Case 1 To 8
                  iX1d_b = iadd - 1
                  ConversionD_B
                  sInstruccionesOperandos(ejec1, 5) = "11001" & iMatrizB(5) & iMatrizB(6) & iMatrizB(7)
                  Exit For
                Case 9 To 10
                  sInstruccionesOperandos(ejec1, 5) = "1100011" & (iadd - 9)
                  Exit For
              End Select
            End If
          Next iadd
          If kmov = 0 Then
            sInstruccionesOperandos(ejec1, 8) = "2"
            sInstruccionesOperandos(ejec1, 9) = "1"
            sInstruccionesOperandos(ejec1, 5) = "11000101"
            LineaLeida1 = sInstruccionesOperandos(ejec1, 3)
            iLongitud1 = Len(LineaLeida1)
            Num_Data
            iX1d_b = iNum_Data
            ConversionD_B
            For Add = 0 To 7
              sInstruccionesOperandos(ejec1, 6) = sInstruccionesOperandos(ejec1, 6) & iMatrizB(Add)
            Next Add
          End If
        End If
        'xch
        Exit For
      Case 28
'       InstruccionADDC
        Y = 0
        kmov = 0
        UbiBancoRnData
        If sInstruccionesOperandos(ejec1, 2) = "A" Then
          For iadd = 1 To 10
            If sInstruccionesOperandos(ejec1, 3) = sMatriz_Operando12(iadd) Then
              kmov = 1
              sInstruccionesOperandos(ejec1, 8) = "1"
              sInstruccionesOperandos(ejec1, 9) = "1"
              Select Case iadd
                Case 1 To 8
                  iX1d_b = iadd - 1
                  ConversionD_B
                  sInstruccionesOperandos(ejec1, 5) = "00111" & iMatrizB(5) & iMatrizB(6) & iMatrizB(7)
                  Exit For
```

```
                Case 9 To 10
                  sInstruccionesOperandos(ejec1, 5) = "0011011" & (iadd - 9)
                  Exit For
              End Select
            End If
          Next iadd
          If kmov = 0 Then
            sInstruccionesOperandos(ejec1, 8) = "2"
            sInstruccionesOperandos(ejec1, 9) = "1"
            LineaLeida1 = sInstruccionesOperandos(ejec1, 3)
            iLongitud1 = Len(LineaLeida1)
            Y = Y + 1
            LetrasLeidas = Mid(LineaLeida1, Y, 1)
            LetrasLeidas = UCase(LetrasLeidas)
            If LetrasLeidas = "#" Then
              sInstruccionesOperandos(ejec1, 5) = "00110100"
              Num_Data
              iX1d_b = iNum_Data
              ConversionD_B
              For Add = 0 To 7
                sInstruccionesOperandos(ejec1, 6) = sInstruccionesOperandos(ejec1, 6) & iMatrizB(Add)
              Next Add
            Else
              sInstruccionesOperandos(ejec1, 5) = "00110101"
              Y = Y - 1
              Num_Data
              iX1d_b = iNum_Data
              ConversionD_B
              For Add = 0 To 7
                sInstruccionesOperandos(ejec1, 6) = sInstruccionesOperandos(ejec1, 6) & iMatrizB(Add)
              Next Add
            End If
          End If
        End If
        'addc
        Exit For
      Case 29
        InstruccionAJMP
        Y = 0
        sInstruccionesOperandos(ejec1, 8) = "2"
        sInstruccionesOperandos(ejec1, 9) = "2"
        For iajmp = 0 To 29
          If sInstruccionesOperandos(ejec1, 2) = sSaltos_Rel(iajmp) Then
            iajmp1 = iSaltos_Rel(iajmp)
            Exit For
          End If
        Next iajmp
        i16Bits1 = iajmp1
        ConversionDPTRD_B
        sInstruccionesOperandos(ejec1, 5) = iBitDPTR(5) & iBitDPTR(6) & iBitDPTR(7) & "00001"
        For Add = 0 To 7
          sInstruccionesOperandos(ejec1, 6) = sInstruccionesOperandos(ejec1, 6) & iBitDPTR(Add + 8)
        Next Add
        'ajmp
        Exit For
      Case 30
        InstruccionCJNE
        Y = 0
        UbiBancoRnData
        sInstruccionesOperandos(ejec1, 8) = "3"
        sInstruccionesOperandos(ejec1, 9) = "2"
        For iadd1 = 0 To 10
          If sInstruccionesOperandos(ejec1, 2) = sMatriz_Operando12(iadd1) Then
          Select Case iadd1
            Case 0
              sInstruccionesOperandos(ejec1, 5) = "10110100"
              LineaLeida1 = sInstruccionesOperandos(ejec1, 3)
              iLongitud1 = Len(LineaLeida1)
              Y = Y + 1
              LetrasLeidas = Mid(LineaLeida1, Y, 1)
              LetrasLeidas = UCase(LetrasLeidas)
```

```
        If LetrasLeidas = "#" Then
          Num_Data
          iX1d_b = iNum_Data
          ConversionD_B
          For Add = 0 To 7
            sInstruccionesOperandos(ejec1, 6) = sInstruccionesOperandos(ejec1, 6) & iMatrizB(Add)
          Next Add
          If iMemoriaRAMintD(224) <> iNum_Data Then
            For icjne = 0 To 29
              If sInstruccionesOperandos(ejec1, 4) = sSaltos_Rel(icjne) Then
                icjne1 = iSaltos_Rel(icjne) - 1
                Exit For
              End If
            Next icjne
          End If
          iX1d_b = icjne1
          ConversionD_B
          For Add = 0 To 7
            sInstruccionesOperandos(ejec1, 7) = sInstruccionesOperandos(ejec1, 7) & iMatrizB(Add)
          Next Add
        Else
          Y = Y - 1
          sInstruccionesOperandos(ejec1, 5) = "10110101"
          Num_Data
          iX1d_b = iNum_Data
          ConversionD_B
          For Add = 0 To 7
            sInstruccionesOperandos(ejec1, 6) = sInstruccionesOperandos(ejec1, 6) & iMatrizB(Add)
          Next Add
          If iMemoriaRAMintD(224) <> iMemoriaRAMintD(iNum_Data) Then
            For icjne = 0 To 29
              If sInstruccionesOperandos(ejec1, 4) = sSaltos_Rel(icjne) Then
                icjne1 = iSaltos_Rel(icjne) - 1
                Exit For
              End If
            Next icjne
          End If
          iX1d_b = icjne1
          ConversionD_B
          For Add = 0 To 7
            sInstruccionesOperandos(ejec1, 7) = sInstruccionesOperandos(ejec1, 7) & iMatrizB(Add)
          Next Add
        End If
        Exit For    'iadd1
      Case 1 To 8
        iX1d_b = iadd1
        ConversionD_B
        sInstruccionesOperandos(ejec1, 5) = "10111" & iMatrizB(5) & iMatrizB(5) & iMatrizB(5)
        LineaLeida1 = sInstruccionesOperandos(ejec1, 3)
        iLongitud1 = Len(LineaLeida1)
        Y = Y + 1
        LetrasLeidas = Mid(LineaLeida1, Y, 1)
        LetrasLeidas = UCase(LetrasLeidas)
        If LetrasLeidas = "#" Then
          Num_Data
          iX1d_b = iNum_Data
          ConversionD_B
          For Add = 0 To 7
            sInstruccionesOperandos(ejec1, 6) = sInstruccionesOperandos(ejec1, 6) & iMatrizB(Add)
          Next Add
          For icjne = 0 To 29
            If sInstruccionesOperandos(ejec1, 4) = sSaltos_Rel(icjne) Then
              icjne1 = iSaltos_Rel(icjne) - 1
            Exit For
            End If
          Next icjne
          iX1d_b = icjne1
          ConversionD_B
          For Add = 0 To 7
            sInstruccionesOperandos(ejec1, 7) = sInstruccionesOperandos(ejec1, 7) & iMatrizB(Add)
          Next Add
```

```
            End If
            Exit For   'iadd1
          Case 9 To 10
            sInstruccionesOperandos(ejec1, 5) = "10110111" & (iadd1 - 9)
            LineaLeida1 = sInstruccionesOperandos(ejec1, 3)
            iLongitud1 = Len(LineaLeida1)
            Y = Y + 1
            LetrasLeidas = Mid(LineaLeida1, Y, 1)
            LetrasLeidas = UCase(LetrasLeidas)
            If LetrasLeidas = "#" Then
              Num_Data
              iX1d_b = iNum_Data
              ConversionD_B
              For Add = 0 To 7
                sInstruccionesOperandos(ejec1, 6) = sInstruccionesOperandos(ejec1, 6) & iMatrizB(Add)
              Next Add
              For icjne = 0 To 29
                If sInstruccionesOperandos(ejec1, 4) = sSaltos_Rel(icjne) Then
                  icjne1 = iSaltos_Rel(icjne) - 1
                Exit For
                End If
              Next icjne
              iX1d_b = icjne1
              ConversionD_B
              For Add = 0 To 7
                sInstruccionesOperandos(ejec1, 7) = sInstruccionesOperandos(ejec1, 7) & iMatrizB(Add)
              Next Add
            End If
            Exit For   'iadd1
        End Select
        End If
      Next iadd1
      'cjne
      Exit For
    Case 31
      InstruccionDJNZ
      Y = 0
      kmov = 0
      UbiBancoRnData
      sInstruccionesOperandos(ejec1, 9) = "2"
      For iadd1 = 1 To 8
        If sInstruccionesOperandos(ejec1, 2) = sMatriz_Operando12(iadd1) Then
          kmov = 1
          sInstruccionesOperandos(ejec1, 8) = "2"
          iX1d_b = iadd1 - 1
          ConversionD_B
          sInstruccionesOperandos(ejec1, 5) = "11011" & iMatrizB(5) & iMatrizB(6) & iMatrizB(7)
          For icjne = 0 To 29
            If sInstruccionesOperandos(ejec1, 3) = sSaltos_Rel(icjne) Then
              icjne1 = iSaltos_Rel(icjne) - 1
              Exit For
            End If
          Next icjne
          iX1d_b = icjne1
          ConversionD_B
          For Add = 0 To 7
            sInstruccionesOperandos(ejec1, 6) = sInstruccionesOperandos(ejec1, 6) & iMatrizB(Add)
          Next Add
        End If
        Exit For   'iadd1
      Next iadd1
      If kmov = 0 Then
        sInstruccionesOperandos(ejec1, 8) = "3"
        sInstruccionesOperandos(ejec1, 5) = "11010101"
        LineaLeida1 = sInstruccionesOperandos(ejec1, 2)
        iLongitud1 = Len(LineaLeida1)
        Num_Data
        iX1d_b = iNum_Data
        ConversionD_B
        For Add = 0 To 7
          sInstruccionesOperandos(ejec1, 6) = sInstruccionesOperandos(ejec1, 6) & iMatrizB(Add)
```

```
            Next Add
            For icjne = 0 To 29
              If sInstruccionesOperandos(ejec1, 3) = sSaltos_Rel(icjne) Then
                icjne1 = iSaltos_Rel(icjne) - 1
                Exit For
              End If
            Next icjne
            iX1d_b = icjne1
            ConversionD_B
            For Add = 0 To 7
              sInstruccionesOperandos(ejec1, 7) = sInstruccionesOperandos(ejec1, 7) & iMatrizB(Add)
            Next Add
          End If
          'djnz
          Exit For
        Case 32
'         InstruccionLJMP
          Y = 0
          sInstruccionesOperandos(ejec1, 8) = "3"
          sInstruccionesOperandos(ejec1, 9) = "2"
          sInstruccionesOperandos(ejec1, 5) = "00000010"
          For iljmp = 0 To 29
            If sInstruccionesOperandos(ejec1, 2) = sSaltos_Rel(iljmp) Then
              iljmp1 = iSaltos_Rel(iljmp) - 1
              Exit For
            End If
          Next iljmp
          i16Bits1 = iljmp1
          ConversionDPTRD_B
          For Add = 0 To 7
            sInstruccionesOperandos(ejec1, 6) = sInstruccionesOperandos(ejec1, 6) & iBitDPTR(Add)
            sInstruccionesOperandos(ejec1, 7) = sInstruccionesOperandos(ejec1, 7) & iBitDPTR(Add + 8)
          Next Add
          'ljmp
          Exit For
        Case 33
'         InstruccionMOVC
          Y = 0
          sInstruccionesOperandos(ejec1, 8) = "1"
          sInstruccionesOperandos(ejec1, 9) = "2"
          If sInstruccionesOperandos(ejec1, 2) = "A" Then
            If sInstruccionesOperandos(ejec1, 3) = "@A+DPTR" Then
              sInstruccionesOperandos(ejec1, 5) = "10010011"
            ElseIf sInstruccionesOperandos(ejec1, 3) = "@A+PC" Then
              sInstruccionesOperandos(ejec1, 5) = "10000011"
            End If
          End If
          'movc
          Exit For
        Case 34
'         InstruccionMOVX
          Y = 0
          UbiBancoRnData
          sInstruccionesOperandos(ejec1, 8) = "1"
          sInstruccionesOperandos(ejec1, 9) = "2"
          For iadd1 = 0 To 12
            If sInstruccionesOperandos(ejec1, 2) = sMatriz_Operando12(iadd1) Then
              Select Case iadd1
                Case 0
                  For iadd = 9 To 12
                    If sInstruccionesOperandos(ejec1, 3) = sMatriz_Operando12(iadd) Then
                      Select Case iadd
                        Case 9 To 10
                          sInstruccionesOperandos(ejec1, 5) = "1110001" & (iadd - 9)
                          Exit For   'iadd
                        Case 12
                          sInstruccionesOperandos(ejec1, 5) = "11100000"
                          Exit For   'iadd
                      End Select
                    End If
                  Next iadd
```

```
                Exit For    'iadd1
              Case 9 To 10
                If sInstruccionesOperandos(ejec1, 3) = "A" Then
                  sInstruccionesOperandos(ejec1, 5) = "1111001" & (iadd1 - 9)
                End If
                Exit For    'iadd1
              Case 12
                If sInstruccionesOperandos(ejec1, 3) = "A" Then
                  sInstruccionesOperandos(ejec1, 5) = "11110000"
                End If
                Exit For    'iadd1
            End Select
          End If
        Next iadd1
        'movx
        Exit For
      Case 35
        InstruccionPUSH
        Y = 0
        sInstruccionesOperandos(ejec1, 8) = "2"
        sInstruccionesOperandos(ejec1, 9) = "2"
        sInstruccionesOperandos(ejec1, 5) = "11000000"
        LineaLeida1 = sInstruccionesOperandos(ejec1, 2)
        iLongitud1 = Len(LineaLeida1)
        Num_Data
        If sInstruccionesOperandos(ejec1, 2) = "A" Then
          iX1d_b = 224
        Else
          iX1d_b = iNum_Data
        End If
        ConversionD_B
        For Add = 0 To 7
          sInstruccionesOperandos(ejec1, 6) = sInstruccionesOperandos(ejec1, 6) & iMatrizB(Add)
        Next Add
        'push
        Exit For
      Case 36
        InstruccionRETI
        Y = 0
        sInstruccionesOperandos(ejec1, 8) = "1"
        sInstruccionesOperandos(ejec1, 9) = "2"
        sInstruccionesOperandos(ejec1, 5) = "00110010"
        'reti
        Exit For
      Case 37
        InstruccionSUBB
        Y = 0
        kmov = 0
        UbiBancoRnData
        sInstruccionesOperandos(ejec1, 9) = "1"
        If sInstruccionesOperandos(ejec1, 2) = "A" Then
          For iadd = 1 To 10
            If sInstruccionesOperandos(ejec1, 3) = sMatriz_Operando12(iadd) Then
              kmov = 1
              sInstruccionesOperandos(ejec1, 8) = "1"
              Select Case iadd
                Case 1 To 8
                  iX1d_b = iadd - 1
                  ConversionD_B
                  sInstruccionesOperandos(ejec1, 5) = "10011" & iMatrizB(5) & iMatrizB(6) & iMatrizB(7)
                  Exit For
                Case 9 To 10
                  sInstruccionesOperandos(ejec1, 5) = "1001011" & (iadd - 9)
                  Exit For
              End Select
            End If
          Next iadd
          If kmov = 0 Then
            sInstruccionesOperandos(ejec1, 8) = "2"
            LineaLeida1 = sInstruccionesOperandos(ejec1, 3)
            iLongitud1 = Len(LineaLeida1)
```

```
            Y = Y + 1
            LetrasLeidas = Mid(LineaLeida1, Y, 1)
            LetrasLeidas = UCase(LetrasLeidas)
            If LetrasLeidas = "#" Then
              sInstruccionesOperandos(ejec1, 5) = "10010100"
              Num_Data
              iX1d_b = iNum_Data
              ConversionD_B
              For Add = 0 To 7
                sInstruccionesOperandos(ejec1, 6) = sInstruccionesOperandos(ejec1, 6) & iMatrizB(Add)
              Next Add
            Else
              Y = Y - 1
              sInstruccionesOperandos(ejec1, 5) = "10010101"
              Num_Data
              iX1d_b = iNum_Data
              ConversionD_B
              For Add = 0 To 7
                sInstruccionesOperandos(ejec1, 6) = sInstruccionesOperandos(ejec1, 6) & iMatrizB(Add)
              Next Add
            End If
          End If
        End If
        'subb
        Exit For
      Case 38
        InstruccionSWAP
        Y = 0
        sInstruccionesOperandos(ejec1, 8) = "1"
        sInstruccionesOperandos(ejec1, 9) = "1"
        sInstruccionesOperandos(ejec1, 5) = "11000100"
        'swap
        Exit For
      Case 39
        InstruccionSETB
        Y = 0
        kmov = 0
        sInstruccionesOperandos(ejec1, 9) = "1"
        If sInstruccionesOperandos(ejec1, 2) = "C" Then
          kmov = 1
          sInstruccionesOperandos(ejec1, 8) = "1"
          sInstruccionesOperandos(ejec1, 5) = "11010011"
        End If
        If kmov = 0 Then
          sInstruccionesOperandos(ejec1, 8) = "2"
          sInstruccionesOperandos(ejec1, 5) = "11010010"
          iOperando = 2
          Ubicacion_Bit
          If iNum_Byte >= 32 And iNum_Byte <= 47 Then
            iX1d_b = (iNum_Byte - 32) * 8 - iNum_Bit + 7
          Else
            iX1d_b = iNum_Byte - iNum_Bit + 7
          End If
          ConversionD_B
          For Add = 0 To 7
            sInstruccionesOperandos(ejec1, 6) = sInstruccionesOperandos(ejec1, 6) & iMatrizB(Add)
          Next Add
        End If
        'setb
        Exit For
      Case 40
        InstruccionSJMP
          Y = 0
          sInstruccionesOperandos(ejec1, 8) = "2"
          sInstruccionesOperandos(ejec1, 9) = "2"
          sInstruccionesOperandos(ejec1, 5) = "10000000"
          For isjmp = 0 To 29
            If sInstruccionesOperandos(ejec1, 2) = sSaltos_Rel(isjmp) Then
              isjmp1 = iSaltos_Rel(isjmp) - 1
              Exit For
            End If
```

```
            Next isjmp
            iX1d_b = isjmp1
            ConversionD_B
            For Add = 0 To 7
              sInstruccionesOperandos(ejec1, 6) = sInstruccionesOperandos(ejec1, 6) & iMatrizB(Add)
            Next Add
          'sjmp
          Exit For
        Case 41
'         InstruccionXCHD
          Y = 0
          UbiBancoRnData
          sInstruccionesOperandos(ejec1, 8) = "1"
          sInstruccionesOperandos(ejec1, 9) = "1"
          If sInstruccionesOperandos(ejec1, 2) = "A" Then
            For iadd = 9 To 10
              If sInstruccionesOperandos(ejec1, 3) = sMatriz_Operando12(iadd) Then
                sInstruccionesOperandos(ejec1, 5) = "1101011" & (iadd - 9)
                Exit For
              End If
            Next iadd
          End If
          'xchd
          Exit For
        Case 42
'         InstruccionACALL
          Y = 0
          sInstruccionesOperandos(ejec1, 8) = "2"
          sInstruccionesOperandos(ejec1, 9) = "2"
          For ilcall = 0 To 29
            If sInstruccionesOperandos(ejec1, 2) = sSaltos_Rel(ilcall) Then
              ilcall1 = iSaltos_Rel(ilcall) - 1
              i16Bits1 = ilcall1
              ConversionDPTRD_B
              sInstruccionesOperandos(ejec1, 5) = iBitDPTR(5) & iBitDPTR(6) & iBitDPTR(7) & "10001"
              For Add = 0 To 7
                sInstruccionesOperandos(ejec1, 6) = sInstruccionesOperandos(ejec1, 6) & iBitDPTR(Add + 8)
              Next Add
              Exit For
            End If
          Next ilcall
          'acall
          Exit For
        Case 43
'         InstruccionLCALL
          Y = 0
          sInstruccionesOperandos(ejec1, 8) = "3"
          sInstruccionesOperandos(ejec1, 9) = "2"
          sInstruccionesOperandos(ejec1, 5) = "00010010"
          For ilcall = 0 To 29
            If sInstruccionesOperandos(ejec1, 2) = sSaltos_Rel(ilcall) Then
              ilcall1 = iSaltos_Rel(ilcall) - 1
              i16Bits1 = ilcall1
              ConversionDPTRD_B
              For Add = 0 To 7
                sInstruccionesOperandos(ejec1, 6) = sInstruccionesOperandos(ejec1, 6) & iBitDPTR(Add)
                sInstruccionesOperandos(ejec1, 7) = sInstruccionesOperandos(ejec1, 7) & iBitDPTR(Add + 8)
              Next Add
              Exit For
            End If
          Next ilcall
          'lcall
          Exit For
      End Select
    End If
      'Manejamos el PC en la columna 25 de la matriz
      'sInstruccionesOperandos
      If ejec1 = 0 Then
        sInstruccionesOperandos(ejec1, 25) = "0"
      Else
```

```
            sInstruccionesOperandos(ejec1, 25) = Str(Val(sInstruccionesOperandos(ejec1 - 1, 25)) +
Val(sInstruccionesOperandos(ejec1 - 1, 8)))
        End If
    Next ejec2
Wend
SaltosBytesVerdaderos
'Generamos el código hexadecimal de ciertos contenidos
'binarios de la matriz sInstruccionesOperandos(n, )
For j = 0 To iMatriz
    For j1 = 5 To 7
        LineaLeida1 = sInstruccionesOperandos(j, j1)
        For j2 = 1 To 8
            LetrasLeidas = Mid(LineaLeida1, j2, 1)
            iValorB(j2 - 1) = Val(LetrasLeidas)
        Next j2
        LineaLeida1 = ""
        ConversionB_D
        iX = iValorD
        ConversionD_H
        sInstruccionesOperandos(j, j1 + 11) = sValorH
    Next j1
Next j
'Para direccionamiento P2
For j = 0 To iMatriz
    For j1 = 0 To 2
        LineaLeida1 = sInstruccionesOperandos(j, j1 * 2 + 10)
        For j2 = 0 To 4
            iValorB(j2) = 0
        Next j2
        For j2 = 6 To 8
            LetrasLeidas = Mid(LineaLeida1, j2 - 5, 1)
            iValorB(j2 - 1) = Val(LetrasLeidas)
        Next j2
        LineaLeida1 = ""
        ConversionB_D
        iX = iValorD
        ConversionD_H
        sInstruccionesOperandos(j, j1 * 2 + 19) = Mid(sValorH, 2, 1)
    Next j1
Next j
'Para direccionamiento P0
For j = 0 To iMatriz
    For j1 = 0 To 2
        LineaLeida1 = sInstruccionesOperandos(j, j1 * 2 + 11)
        For j2 = 1 To 8
            LetrasLeidas = Mid(LineaLeida1, j2, 1)
            iValorB(j2 - 1) = Val(LetrasLeidas)
        Next j2
        LineaLeida1 = ""
        ConversionB_D
        iX = iValorD
        ConversionD_H
        sInstruccionesOperandos(j, j1 * 2 + 20) = sValorH
    Next j1
Next j
ejec1 = -1
End Sub

Private Sub cmdStep_Click()
Frecuencia_Simulación
Step = Step + 1
If Step <= 1 Then
    ejec1 = -1
End If
ejec1 = ejec1 + 1
PC = ejec1
    For ejec2 = 0 To 43
        If sInstruccionesOperandos(ejec1, 1) = sMatriz_Opcode(ejec2) Then
            Select Case ejec2
                Case 0
                    InstruccionDA   'Esta subrrutina permite
```

```
                'realizar el ajuste decimal
                'del acumulador
    Exit For
  Case 1
    InstruccionJC
    'jc
    Exit For
  Case 2
    InstruccionJB
    'jb
    Exit For
  Case 3
    InstruccionJZ
    'jz
    Exit For
  Case 4
    InstruccionRL
    'rl
    Exit For
  Case 5
    InstruccionRR
    'rr
    Exit For
  Case 6
    InstruccionADD
    'add
    Exit For
  Case 7
    InstruccionANL
    'anl
    Exit For
  Case 8
    InstruccionCLR
    'clr
    Exit For
  Case 9
    InstruccionCPL
    'cpl
    Exit For
  Case 10
    InstruccionDIV
    'div
    Exit For
  Case 11
    InstruccionDEC
    'dec
    Exit For
  Case 12
    InstruccionINC
    'inc
    Exit For
  Case 13
    InstruccionJNC
    'jnc
    Exit For
  Case 14
    InstruccionJNB
    'jnb
    Exit For
  Case 15
    InstruccionJBC
    'jbc
    Exit For
  Case 16
    InstruccionJMP
    'jmp
    Exit For
  Case 17
    InstruccionJNZ
    'jnz
    Exit For
```

```
Case 18
   InstruccionMUL
   'mul
   Exit For
Case 19
   InstruccionMOV
   'mov
   Exit For
Case 20
   InstruccionNOP
   'nop
   Exit For
Case 21
   InstruccionORL
   'orl
   Exit For
Case 22
   InstruccionPOP
   'pop
   Exit For
Case 23
   InstruccionRLC
   'rlc
   Exit For
Case 24
   InstruccionRRC
   'rrc
   Exit For
Case 25
   InstruccionRET
   'ret
   Exit For
Case 26
   InstruccionXRL
   'xrl
   Exit For
Case 27
   InstruccionXCH
   'xch
   Exit For
Case 28
   InstruccionADDC
   Exit For
   'addc
Case 29
   InstruccionAJMP
   'ajmp
   Exit For
Case 30
   InstruccionCJNE
   'cjne
   Exit For
Case 31
   InstruccionDJNZ
   'djnz
   Exit For
Case 32
   InstruccionLJMP
   'ljmp
   Exit For
Case 33
   InstruccionMOVC
   'movc
   Exit For
Case 34
   InstruccionMOVX
   'movx
   Exit For
Case 35
   InstruccionPUSH
   'push
```

```
            Exit For
          Case 36
            InstruccionRETI
            'reti
            Exit For
          Case 37
            InstruccionSUBB
            'subb
            Exit For
          Case 38
            InstruccionSWAP
            'swap
            Exit For
          Case 39
            InstruccionSETB
            'setb
            Exit For
          Case 40
            InstruccionSJMP
            'sjmp
            Exit For
          Case 41
            InstruccionXCHD
            'xchd
            Exit For
          Case 42
            InstruccionACALL
            'acall
            Exit For
          Case 43
            InstruccionLCALL
            'lcall
            Exit For
        End Select
      End If
    Next ejec2
End Sub

Private Sub Command2_Click()
frmMemorias.Show
End Sub

Private Sub Command3_Click()
'Boton para pruebas
grdInstrucciones.Rows = iMatriz + 2
For i = 1 To 8
   grdInstrucciones.ColWidth(i) = 700
Next
For i = 9 To 10
   grdInstrucciones.ColWidth(i) = 120
Next
For i = 11 To 16
   grdInstrucciones.ColWidth(i) = 700
Next
For i = 1 To iMatriz + 1
   grdInstrucciones.Row = i
   For j = 1 To 26
      grdInstrucciones.Col = j
      grdInstrucciones.Text = sInstruccionesOperandos(i - 1, j - 1)
   Next j
Next i
For i = 1 To iMatriz + 1
   grdInstrucciones.Row = i
   grdInstrucciones.Col = 0
   grdInstrucciones.Text = i
Next i
End Sub

Private Sub Form_Load()
sMatriz_Opcode(0) = "DA"
sMatriz_Opcode(1) = "JC"
```

```
sMatriz_Opcode(2) = "JB"
sMatriz_Opcode(3) = "JZ"
sMatriz_Opcode(4) = "RL"
sMatriz_Opcode(5) = "RR"
sMatriz_Opcode(6) = "ADD"
sMatriz_Opcode(7) = "ANL"
sMatriz_Opcode(8) = "CLR"
sMatriz_Opcode(9) = "CPL"
sMatriz_Opcode(10) = "DIV"
sMatriz_Opcode(11) = "DEC"
sMatriz_Opcode(12) = "INC"
sMatriz_Opcode(13) = "JNC"
sMatriz_Opcode(14) = "JNB"
sMatriz_Opcode(15) = "JBC"
sMatriz_Opcode(16) = "JMP"
sMatriz_Opcode(17) = "JNZ"
sMatriz_Opcode(18) = "MUL"
sMatriz_Opcode(19) = "MOV"
sMatriz_Opcode(20) = "NOP"
sMatriz_Opcode(21) = "ORL"
sMatriz_Opcode(22) = "POP"
sMatriz_Opcode(23) = "RLC"
sMatriz_Opcode(24) = "RRC"
sMatriz_Opcode(25) = "RET"
sMatriz_Opcode(26) = "XRL"
sMatriz_Opcode(27) = "XCH"
sMatriz_Opcode(28) = "ADDC"
sMatriz_Opcode(29) = "AJMP"
sMatriz_Opcode(30) = "CJNE"
sMatriz_Opcode(31) = "DJNZ"
sMatriz_Opcode(32) = "LJMP"
sMatriz_Opcode(33) = "MOVC"
sMatriz_Opcode(34) = "MOVX"
sMatriz_Opcode(35) = "PUSH"
sMatriz_Opcode(36) = "RETI"
sMatriz_Opcode(37) = "SUBB"
sMatriz_Opcode(38) = "SWAP"
sMatriz_Opcode(39) = "SETB"
sMatriz_Opcode(40) = "SJMP"
sMatriz_Opcode(41) = "XCHD"
sMatriz_Opcode(42) = "ACALL"
sMatriz_Opcode(43) = "LCALL"
sMatriz_Operando12(0) = "A"
sMatriz_Operando12(1) = "R0"
sMatriz_Operando12(2) = "R1"
sMatriz_Operando12(3) = "R2"
sMatriz_Operando12(4) = "R3"
sMatriz_Operando12(5) = "R4"
sMatriz_Operando12(6) = "R5"
sMatriz_Operando12(7) = "R6"
sMatriz_Operando12(8) = "R7"
sMatriz_Operando12(9) = "@R0"
sMatriz_Operando12(10) = "@R1"
sMatriz_Operando12(11) = "DPTR"
sMatriz_Operando12(12) = "@DPTR"
sMatriz_Operando12(13) = "C"
sMatriz_Operando12(14) = "@A+DPTR"
sMatriz_Operando12(15) = "@A+PC"
sSFR_Operandos(0, 0) = "B"
sSFR_Operandos(0, 1) = "240"
sSFR_Operandos(1, 0) = "PSW"
sSFR_Operandos(1, 1) = "208"
sSFR_Operandos(2, 0) = "SP"
sSFR_Operandos(2, 1) = "129"
sSFR_Operandos(3, 0) = "DPL"
sSFR_Operandos(3, 1) = "130"
sSFR_Operandos(4, 0) = "DPH"
sSFR_Operandos(4, 1) = "131"
sSFR_Operandos(5, 0) = "P0"
sSFR_Operandos(5, 1) = "128"
sSFR_Operandos(6, 0) = "P1"
```

```
sSFR_Operandos(6, 1) = "144"
sSFR_Operandos(7, 0) = "P2"
sSFR_Operandos(7, 1) = "160"
sSFR_Operandos(8, 0) = "P3"
sSFR_Operandos(8, 1) = "176"
sSFR_Operandos(9, 0) = "IP"
sSFR_Operandos(9, 1) = "184"
sSFR_Operandos(10, 0) = "IE"
sSFR_Operandos(10, 1) = "168"
sSFR_Operandos(11, 0) = "TMOD"
sSFR_Operandos(11, 1) = "137"
sSFR_Operandos(12, 0) = "TCON"
sSFR_Operandos(12, 1) = "136"
sSFR_Operandos(13, 0) = "TH0"
sSFR_Operandos(13, 1) = "140"
sSFR_Operandos(14, 0) = "TL0"
sSFR_Operandos(14, 1) = "138"
sSFR_Operandos(15, 0) = "TH1"
sSFR_Operandos(15, 1) = "141"
sSFR_Operandos(16, 0) = "TL1"
sSFR_Operandos(16, 1) = "139"
sSFR_Operandos(17, 0) = "SCON"
sSFR_Operandos(17, 1) = "152"
sSFR_Operandos(18, 0) = "SBUF"
sSFR_Operandos(18, 1) = "153"
sSFR_Operandos(19, 0) = "PCON"
sSFR_Operandos(19, 1) = "135"
sSFR_Bits(0, 0) = "RXD"
sSFR_Bits(0, 1) = "176.0"
sSFR_Bits(1, 0) = "TXD"
sSFR_Bits(1, 1) = "176.1"
sSFR_Bits(2, 0) = "INT0"
sSFR_Bits(2, 1) = "176.2"
sSFR_Bits(3, 0) = "INT1"
sSFR_Bits(3, 1) = "176.3"
sSFR_Bits(4, 0) = "T0"
sSFR_Bits(4, 1) = "176.4"
sSFR_Bits(5, 0) = "T1"
sSFR_Bits(5, 1) = "176.5"
sSFR_Bits(6, 0) = "WR"
sSFR_Bits(6, 1) = "176.6"
sSFR_Bits(7, 0) = "RD"
sSFR_Bits(7, 1) = "176.7"
sSFR_Bits(8, 0) = "IT0"
sSFR_Bits(8, 1) = "136.0"
sSFR_Bits(9, 0) = "IE0"
sSFR_Bits(9, 1) = "136.1"
sSFR_Bits(10, 0) = "IT1"
sSFR_Bits(10, 1) = "136.2"
sSFR_Bits(11, 0) = "IE1"
sSFR_Bits(11, 1) = "136.3"
sSFR_Bits(12, 0) = "TR0"
sSFR_Bits(12, 1) = "136.4"
sSFR_Bits(13, 0) = "TF0"
sSFR_Bits(13, 1) = "136.5"
sSFR_Bits(14, 0) = "TR1"
sSFR_Bits(14, 1) = "136.6"
sSFR_Bits(15, 0) = "TF1"
sSFR_Bits(15, 1) = "136.7"
sSFR_Bits(16, 0) = "RI"
sSFR_Bits(16, 1) = "152.0"
sSFR_Bits(17, 0) = "TI"
sSFR_Bits(17, 1) = "152.1"
sSFR_Bits(18, 0) = "RB8"
sSFR_Bits(18, 1) = "152.2"
sSFR_Bits(19, 0) = "TB8"
sSFR_Bits(19, 1) = "152.3"
sSFR_Bits(20, 0) = "REN"
sSFR_Bits(20, 1) = "152.4"
sSFR_Bits(21, 0) = "SM2"
sSFR_Bits(21, 1) = "152.5"
```

```
sSFR_Bits(22, 0) = "SM1"
sSFR_Bits(22, 1) = "152.6"
sSFR_Bits(23, 0) = "SM0"
sSFR_Bits(23, 1) = "152.7"
sSFR_Bits(24, 0) = "IDL"
sSFR_Bits(24, 1) = "135.0"
sSFR_Bits(25, 0) = "PD"
sSFR_Bits(25, 1) = "135.1"
sSFR_Bits(26, 0) = "GF0"
sSFR_Bits(26, 1) = "135.2"
sSFR_Bits(27, 0) = "GF1"
sSFR_Bits(27, 1) = "135.3"
sSFR_Bits(28, 0) = "SMOD"
sSFR_Bits(28, 1) = "135.7"
sSFR_Bits(29, 0) = "ACC.0"
sSFR_Bits(29, 1) = "224.0"
sSFR_Bits(30, 0) = "ACC.1"
sSFR_Bits(30, 1) = "224.1"
sSFR_Bits(31, 0) = "ACC.2"
sSFR_Bits(31, 1) = "224.2"
sSFR_Bits(32, 0) = "ACC.3"
sSFR_Bits(32, 1) = "224.3"
sSFR_Bits(33, 0) = "ACC.4"
sSFR_Bits(33, 1) = "224.4"
sSFR_Bits(34, 0) = "ACC.5"
sSFR_Bits(34, 1) = "224.5"
sSFR_Bits(35, 0) = "ACC.6"
sSFR_Bits(35, 1) = "224.6"
sSFR_Bits(36, 0) = "ACC.7"
sSFR_Bits(36, 1) = "224.7"
sSFR_Bits(37, 0) = "P"
sSFR_Bits(37, 1) = "208.0"
sSFR_Bits(38, 0) = "OV"
sSFR_Bits(38, 1) = "208.2"
sSFR_Bits(39, 0) = "RS0"
sSFR_Bits(39, 1) = "208.3"
sSFR_Bits(40, 0) = "RS1"
sSFR_Bits(40, 1) = "208.4"
sSFR_Bits(41, 0) = "F0"
sSFR_Bits(41, 1) = "208.5"
sSFR_Bits(42, 0) = "AC"
sSFR_Bits(42, 1) = "208.6"
sSFR_Bits(43, 0) = "EX0"
sSFR_Bits(43, 1) = "168.0"
sSFR_Bits(44, 0) = "ET0"
sSFR_Bits(44, 1) = "168.1"
sSFR_Bits(45, 0) = "EX1"
sSFR_Bits(45, 1) = "168.2"
sSFR_Bits(46, 0) = "ET1"
sSFR_Bits(46, 1) = "168.3"
sSFR_Bits(47, 0) = "ES"
sSFR_Bits(47, 1) = "168.4"
sSFR_Bits(48, 0) = "ET2"
sSFR_Bits(48, 1) = "168.5"
sSFR_Bits(49, 0) = "EA"
sSFR_Bits(49, 1) = "168.7"
sSFR_Bits(50, 0) = "PX0"
sSFR_Bits(50, 1) = "184.0"
sSFR_Bits(51, 0) = "PT0"
sSFR_Bits(51, 1) = "184.1"
sSFR_Bits(52, 0) = "PX1"
sSFR_Bits(52, 1) = "184.2"
sSFR_Bits(53, 0) = "PT1"
sSFR_Bits(53, 1) = "184.3"
sSFR_Bits(54, 0) = "PS"
sSFR_Bits(54, 1) = "184.4"
sSFR_Bits(55, 0) = "PT2"
sSFR_Bits(55, 1) = "184.5"
sSaltosVerdaderos(0) = "JB"
sSaltosVerdaderos(1) = "JC"
sSaltosVerdaderos(2) = "JZ"
```

```
sSaltosVerdaderos(3) = "JBC"
sSaltosVerdaderos(4) = "JNB"
sSaltosVerdaderos(5) = "JNC"
sSaltosVerdaderos(6) = "JNZ"
sSaltosVerdaderos(7) = "AJMP"
sSaltosVerdaderos(8) = "CJNE"
sSaltosVerdaderos(9) = "DJNZ"
sSaltosVerdaderos(10) = "LJMP"
sSaltosVerdaderos(11) = "SJMP"
sSaltosVerdaderos(12) = "ACALL"
sSaltosVerdaderos(13) = "LCALL"
'Las siguientes instrucciones permiten cargar
'los valores iniciales de la Memoria RAM
For i = 0 To 127
   For j = 0 To 7
      iMemoriaRAMint(i, j) = 0
   Next j
Next i
For i = 0 To 3  'Para cargar 1 en los porticos de entrada salida
   For j = 0 To 7
      iMemoriaRAMint(128 + i * 16, j) = 1
   Next j
Next i
For j = 0 To 4 'Carga el Stack Pointer SP
   iMemoriaRAMint(129, j) = 0
Next j
For j = 5 To 7
   iMemoriaRAMint(129, j) = 1
Next j
For i = 130 To 134 'Carga DPL, DPH .....y otros registros
   For j = 0 To 7
      iMemoriaRAMint(i, j) = 0
   Next j
Next i
iMemoriaRAMint(135, 0) = 0 'El resto de Bits de PCON son indeterminados
For i = 136 To 143  'Carga TCON, TMOD, TLO,TL1, TH0,TH1 y o tros registros
   For j = 0 To 7
      iMemoriaRAMint(i, j) = 0
   Next j
Next i
For i = 145 To 152 'Carga SCON y otros registros
   For j = 0 To 7
      iMemoriaRAMint(i, j) = 0
   Next j
Next i
'SBUF fila 153 indeterminado
For i = 154 To 159 'Carga otros registros
   For j = 0 To 7
      iMemoriaRAMint(i, j) = 0
   Next j
Next i
For i = 161 To 167 'Carga otros registros
   For j = 0 To 7
      iMemoriaRAMint(i, j) = 0
   Next j
Next i
iMemoriaRAMint(168, 0) = 0 'Carga el registro IE
For j = 3 To 7
   iMemoriaRAMint(168, j) = 0
Next j
For i = 169 To 175 'Carga otros registros
   For j = 0 To 7
      iMemoriaRAMint(i, j) = 0
   Next j
Next i
For i = 177 To 183 'Carga otros registros
   For j = 0 To 7
      iMemoriaRAMint(i, j) = 0
   Next j
Next i
For j = 3 To 7 'Carga el registro IP
```

```
    iMemoriaRAMint(184, j) = 0
  Next j
  For i = 185 To 255
    For j = 0 To 7
      iMemoriaRAMint(i, j) = 0
    Next j
  Next i
  'Las siguientes líneas permiten convertir los datos de la
  'memoria RAM interna de binarios a decimales y hexadecimales
  ActualizacionRAM
  SP = 47
  iMemoriaRAMintD(129) = SP
  frmDiagrama2.Show
  mnuBuffer.Checked = True
  mnuPorticos.Checked = False
  'Inicialización de memoria ROM externa
  iRomc1 = 0
  KbytesRom = 2048
  For ROMext1 = 0 To 2000
    For ROMext2 = 0 To 7
      iMemoriaROMext(ROMext1, ROMext2) = 1
    Next ROMext2
  Next ROMext1
  'Inicialización de Buffer y Puerto P1 Entrada
  sValor_InH_P1 = "00"
  iValor_In_P1 = 0
  For buf = 0 To 7
    saValor_InB_P1(buf) = 0
  Next buf
  'Carga las variables para direccionamientos
  'de memoria RAM externa
  iRam = 1
  iBuffer = 2
  iLatch = 3
  KbytesRam = 2048
  iInterrupcion_EXTI0 = False
  iInterrupcion_Timer0 = False
  iInterrupcion_EXTI1 = False
  iInterrupcion_Timer1 = False
  For irext = 0 To 4095
    For irext1 = 0 To 7
      iMemoriaRAMext(irext, irext1) = 0
    Next irext1
  Next irext
  For isp = 0 To 30
    iMatrizSP(isp) = 0
  Next isp
  iFilasSP = 0
  iSP3 = 0
  iInterrupcion = False
  STimer0 = False
  STimer1 = False
  ActualizacionRAMD_BH
  frmDiagrama1.Visible = False
  End Sub

  Private Sub mnu1_Click()
  mnu1.Checked = True
  mnu2.Checked = False
  mnu3.Checked = False
  mnu4.Checked = False
  End Sub


  Private Sub mnu1KbytesRAM_Click()
  mnu2KbytesRAM.Checked = False
  mnu1KbytesRAM.Checked = True
  mnu4KbytesRAM.Checked = False
  frmDiagrama2.img2KRam.Picture = frmImagenes.imgDib1KRam2.Picture
  frmDiagrama2.img2KRam.Tag = "1K"
  frmDiagrama2.Shape9.Height = 160
```

```
MDIForm1.mnum2KbytesRAM.Checked = False
MDIForm1.mnum1KbytesRAM.Checked = True
MDIForm1.mnum4KbytesRAM.Checked = False
KbytesRam = 1024
If KbytesRom = 2048 Then
   frmDiagrama2.Shape1.Height = 195
ElseIf KbytesRom = 4096 Then
   frmDiagrama2.Shape1.Height = 280
End If
End Sub

Private Sub mnu1KbytesROM_Click()
mnu2KbytesROM.Checked = False
mnu1KbytesROM.Checked = True
mnu4KbytesROM.Checked = False
frmDiagrama2.img2KRom.Left = 8220
frmDiagrama2.img2KRom.Top = 530
frmDiagrama2.img2KRom.Picture = frmImagenes.imgDib1KRom2.Picture
frmDiagrama2.img2KRom.Tag = "1K"
frmDiagrama2.Shape6.Height = 165
frmDiagrama2.Shape1.Height = 110
MDIForm1.mnum2KbytesROM.Checked = False
MDIForm1.mnum1KbytesROM.Checked = True
MDIForm1.mnum4KbytesROM.Checked = False
KbytesRom = 1024
If KbytesRam = 2048 Then
   frmDiagrama2.Shape1.Height = 195
ElseIf KbytesRam = 4096 Then
   frmDiagrama2.Shape1.Height = 280
End If
End Sub

Private Sub mnu2_Click()
mnu1.Checked = False
mnu2.Checked = True
mnu3.Checked = False
mnu4.Checked = False
End Sub

Private Sub mnu2KbytesRAM_Click()
mnu2KbytesRAM.Checked = True
mnu1KbytesRAM.Checked = False
mnu4KbytesRAM.Checked = False
frmDiagrama2.img2KRam.Picture = frmImagenes.imgDib2KRam4.Picture
frmDiagrama2.img2KRam.Tag = "2K"
frmDiagrama2.Shape9.Height = 255
MDIForm1.mnum2KbytesRAM.Checked = True
MDIForm1.mnum1KbytesRAM.Checked = False
MDIForm1.mnum4KbytesRAM.Checked = False
KbytesRam = 2048
If KbytesRam >= KbytesRom Then
   frmDiagrama2.Shape1.Height = 195
Else
   frmDiagrama2.Shape1.Height = 280
End If
End Sub

Private Sub mnu2KbytesROM_Click()
mnu2KbytesROM.Checked = True
mnu1KbytesROM.Checked = False
mnu4KbytesROM.Checked = False
frmDiagrama2.img2KRom.Left = 8190
frmDiagrama2.img2KRom.Top = 495
frmDiagrama2.img2KRom.Picture = frmImagenes.imgDib2KRom4.Picture
frmDiagrama2.img2KRom.Tag = "2K"
frmDiagrama2.Shape6.Height = 255
frmDiagrama2.Shape1.Height = 195
MDIForm1.mnum2KbytesROM.Checked = True
MDIForm1.mnum1KbytesROM.Checked = False
MDIForm1.mnum4KbytesROM.Checked = False
KbytesRom = 2048
```

```
If KbytesRom >= KbytesRam Then
   frmDiagrama2.Shape1.Height = 195
Else
   frmDiagrama2.Shape1.Height = 280
End If
End Sub

Private Sub mnu3_Click()
mnu1.Checked = False
mnu2.Checked = False
mnu3.Checked = True
mnu4.Checked = False
End Sub

Private Sub mnu4_Click()
mnu1.Checked = False
mnu2.Checked = False
mnu3.Checked = False
mnu4.Checked = True
End Sub

Private Sub mnu4KbytesRAM_Click()
mnu4KbytesRAM.Checked = True
mnu1KbytesRAM.Checked = False
mnu2KbytesRAM.Checked = False
frmDiagrama2.img2KRam.Picture = frmImagenes.imgDib4KRam2.Picture
frmDiagrama2.img2KRam.Tag = "4K"
frmDiagrama2.Shape9.Height = 360
MDIForm1.mnum2KbytesRAM.Checked = False
MDIForm1.mnum1KbytesRAM.Checked = False
MDIForm1.mnum4KbytesRAM.Checked = True
KbytesRam = 4096
End Sub

Private Sub mnu4KbytesROM_Click()
mnu4KbytesROM.Checked = True
mnu1KbytesROM.Checked = False
mnu2KbytesROM.Checked = False
frmDiagrama2.img2KRom.Top = 500
frmDiagrama2.img2KRom.Picture = frmImagenes.imgDib4KRom2.Picture
frmDiagrama2.img2KRom.Tag = "4K"
frmDiagrama2.Shape6.Height = 355
frmDiagrama2.Shape1.Height = 280
MDIForm1.mnum2KbytesROM.Checked = False
MDIForm1.mnum1KbytesROM.Checked = False
MDIForm1.mnum4KbytesROM.Checked = True
KbytesRom = 4096
End Sub

Private Sub mnuAbrir_Click()
cmdGenerarMatriz_Click
End Sub

Private Sub mnuBuffer_Click()
Unload frmDiagrama1
frmDiagrama2.Show
mnuBuffer.Checked = True
mnuPorticos.Checked = False
End Sub

Private Sub mnuBuffer541a_Click()
frmBuffer244.Show
End Sub

Private Sub mnuCodigo_Click()
Unload frmCodigo
For codigo1 = 0 To iMatriz
   frmCodigo.cboCodigo.AddItem saMatrizInstrucciones(codigo1)
Next codigo1
   frmCodigo.cboCodigo.ListIndex = ejec1 + 1
frmCodigo.Show
```

```
End Sub

Private Sub mnuDecodificador138_Click()
frmDecodificador138.Show
End Sub

Private Sub mnuDireccionamiento_RAM_Click()
frmDirec_Ram.Show
End Sub

Private Sub mnuDireccionamiento_RAMa_Click()
frmDirec_Ram.Show
End Sub

Private Sub mnuEditarInstrución_Click()
frmEdit_Ins.Show
End Sub

Private Sub mnuLatch377a_Click()
frmLatch373.Show
End Sub

Private Sub mnuMicrocontroiador_Click()
frmMicro.Show
End Sub

Private Sub mnuPorInstruccion_Click()
mnuPorInstruccion.Checked = True
cmdStep_Click
mnuPorInstruccion.Checked = False
End Sub

Private Sub mnuPorInstruccionSA_Click()
mnuPorInstruccionSA.Checked = True
cmdStep_Click
mnuPorInstruccionSA.Checked = False
End Sub

Private Sub mnuPorticos_Click()
If mnuPorticos.Checked = True Then
   mnuPorticos.Checked = False
   frmDiagrama1.Shape11.Visible = False
   frmDiagrama1.Line44.Visible = False
   frmDiagrama1.Label9.Visible = False
   frmDiagrama1.Label7.Visible = False
   frmDiagrama1.Label8.Visible = False
   frmDiagrama1.Image9.Visible = False
   For p1ch = 0 To 7
      frmDiagrama1.imgPuerto_P1(p1ch).Visible = False
   Next p1ch
Else
   mnuPorticos.Checked = True
   frmDiagrama1.Shape11.Visible = True
   frmDiagrama1.Line44.Visible = True
   frmDiagrama1.Label9.Visible = True
   frmDiagrama1.Label7.Visible = True
   frmDiagrama1.Label8.Visible = True
   frmDiagrama1.Image9.Visible = True
   For p1ch = 0 To 7
      frmDiagrama1.imgPuerto_P1(p1ch).Visible = True
   Next p1ch
End If
Unload frmDiagrama2
frmDiagrama1.Show
mnuBuffer.Checked = False
mnuPorticos.Checked = True
End Sub

Private Sub mnuRAMInterna_Click()
frmMemorias.Show
End Sub
```

```
Private Sub mnuROM_Click()
mnuVer_EsquematicoROM_Click
End Sub

Private Sub mnuSalir_Click()
Unload Me
End
End Sub

Private Sub mnuSwitcharP1_Click()
If iBuffer_P1 Then
   frmSwitcheo_P1.Caption = "Configuración Interruptores"
Else
   frmSwitcheo_P1.Caption = "Configuración Interruptores"
End If
frmSwitcheo_P1.Show
frmSwitcheo_P1.txtSwitcheo_P1.SetFocus
End Sub

Private Sub mnuTodo_Click()
mnuTodo.Checked = True
cmdEjecutarPrograma_Click
mnuTodo.Checked = False
End Sub

Private Sub mnuTodoSA_Click()
mnuTodoSA.Checked = True
cmdEjecutarPrograma_Click
mnuTodoSA.Checked = False
End Sub

Private Sub mnuVer_ContenidoRam_Click()
frmRamExterna.Show
End Sub

Private Sub mnuVer_ContenidoRom_Click()
frmMemoriaROM.Show
End Sub

Private Sub mnuVer_Esquematico_Click()
frmDecodificador138.Show
End Sub

Private Sub mnuVer_EsquematicoBuffer541_Click()
frmBuffer244.Show
End Sub

Private Sub mnuVer_EsquematicoLatch377_Click()
frmLatch373.Show
End Sub

Private Sub mnuVer_EsquematicoLatchz_Click()
frmLatch.Show
End Sub

Private Sub mnuVer_EsquematicoMicro_Click()
frmMicro.Show
End Sub

Private Sub mnuVer_EsquematicoRam_Click()
If frmDiagrama2.img2KRam.Tag = "1K" Then
   frm2KRam.Caption = "Memoria Ram Externa 1K"
   frm2KRam.Picture = frmImagenes.imgDib1KRam1.Picture
ElseIf frmDiagrama2.img2KRam.Tag = "2K" Then
   frm2KRam.Caption = "Memoria Ram Externa 2K"
   frm2KRam.Picture = frmImagenes.imgDib2KRam1.Picture
ElseIf frmDiagrama2.img2KRam.Tag = "4K" Then
   frm2KRam.Caption = "Memoria Ram Externa 4K"
   frm2KRam.Picture = frmImagenes.imgDib4KRam1.Picture
End If
```

```
frm2KRam.Show
End Sub

Private Sub mnuVer_EsquematicoROM_Click()
If frmDiagrama2.img2KRom.Tag = "1K" Then
   frm2KRom.Caption = "Memoria Rom 1K"
   frm2KRom.Picture = frmImagenes.imgDib1KRom1.Picture
ElseIf frmDiagrama2.img2KRom.Tag = "2K" Then
   frm2KRom.Caption = "Memoria Rom 2K"
   frm2KRom.Picture = frmImagenes.imgDib2KRom5.Picture
ElseIf frmDiagrama2.img2KRom.Tag = "4K" Then
   frm2KRom.Caption = "Memoria Rom 4K"
   frm2KRom.Picture = frmImagenes.imgDib4KRom1.Picture
End If
frm2KRom.Show
End Sub

Private Sub mnuVer_RAM_Interna_Click()
frmMemorias.Show
End Sub

Private Sub Toolbar1_ButtonClick(ByVal Button As Button)
If Button.Tag = "Abrir" Then
   cmdGenerarMatriz_Click
ElseIf Button.Tag = "PasoaPaso" Then
   mnuPorInstruccion_Click
ElseIf Button.Tag = "Codigo" Then
   mnuCodigo_Click
ElseIf Button.Tag = "Todo" Then
   mnuTodo_Click
ElseIf Button.Tag = "PasoaPasoSA" Then
   mnuPorInstruccionSA_Click
ElseIf Button.Tag = "TodoSA" Then
   mnuTodoSA_Click
End If
End Sub
```

frmCodigoDetallado

```
Private Sub Form_Load()
grdCD.Rows = iMatriz + 2
grdCD.Row = 0
grdCD.Col = 1
grdCD.Text = "ETIQUETA"
grdCD.Col = 2
grdCD.Text = "OPCODE"
grdCD.Col = 3
grdCD.Text = "OPERANDO 1"
grdCD.Col = 4
grdCD.Text = "OPERANDO 2"
grdCD.Col = 5
grdCD.Text = "OPERANDO 3"
grdCD.Col = 6
grdCD.Text = "# BYTES"
grdCD.Col = 7
grdCD.Text = "# C.M"
grdCD.Col = 8
grdCD.Text = "BYTE 1"
grdCD.Col = 9
grdCD.Text = "BYTE 2"
grdCD.Col = 10
grdCD.Text = "BYTE 3"
grdCD.ColAlignment(6) = 2
grdCD.ColAlignment(7) = 2
grdCD.ColAlignment(8) = 2
grdCD.ColAlignment(9) = 2
grdCD.ColAlignment(10) = 2
For i = 1 To 5
   grdCD.ColWidth(i) = 1100
Next
For i = 6 To 10
```

```
    grdCD.ColWidth(i) = 700
  Next
  For i = 1 To iMatriz + 1
    grdCD.Row = i
    For j = 1 To 5
      grdCD.Col = j
      grdCD.Text = sInstruccionesOperandos(i - 1, j - 1)
    Next j
    For j = 6 To 7
      grdCD.Col = j
      grdCD.Text = sInstruccionesOperandos(i - 1, j + 2)
    Next j
    For j = 8 To 10
      grdCD.Col = j
      grdCD.Text = sInstruccionesOperandos(i - 1, j + 8)
    Next j
  Next i
  For i = 1 To iMatriz + 1
    grdCD.Row = i
    grdCD.Col = 0
    grdCD.Text = i
  Next i
  End Sub
```

frmDiagrama1

```
'Declaración  de variables
Dim icolor As Long
Dim icolor2 As Long
Dim icolor3 As Long

Sub ALE_Latch_color1()
frmDiagrama1.Line1.BorderColor = icolor
frmDiagrama1.Line2.BorderColor = icolor
frmDiagrama1.Line3.BorderColor = icolor
End Sub

Sub ALE_Latch_QBcolor1()
frmDiagrama1.Line1.BorderColor = QBColor(12)
frmDiagrama1.Line2.BorderColor = QBColor(12)
frmDiagrama1.Line3.BorderColor = QBColor(12)
End Sub

Sub Byte1_1Ciclo_1()
'Cada fase tendrá una duración de 2*tiem segundos
tiem = Frecuencia_Oscilador
icolor = 16777215   'Line1.BorderColor
icolor2 = 16777215  'Shape1.BackColor
icolor3 = 8421504   'Shape4.BackColor
For icolor1 = 1 To 2
txtDir_P2.Text = sInstruccionesOperandos(ejec1, 19)
txtDirDat_P0.Text = sInstruccionesOperandos(ejec1, 20)
'Segunda fase primer estado, primera fase segundo estado,
label1.Caption = saMatrizInstrucciones(ejec1)
Label2.Caption = "#Bytes = " & sInstruccionesOperandos(ejec1, 8)
Label3.Caption = "#C.Maq = " & sInstruccionesOperandos(ejec1, 9)
label1.Visible = True
Label2.Visible = True
Label3.Visible = True
ALE_Latch_QBcolor1
PSEN_Eprom_QBcolor1
P2_Eprom_QBcolor1
P0_Eprom_color1
frmDiagrama1.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
P2_Eprom_color1
frmDiagrama1.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
P2_Eprom_QBcolor1
P0_Latch_QBcolor1
'El direccionamiento realmente comienza en este estado
```

```
Label4.Caption = "Direccionamiento a ROM"
Label4.Visible = True
Label4.ForeColor = QBColor(4)
txtDir_P2.Visible = True
txtDirDat_P0.Visible = True
imgArrow1.Visible = True
imgArrow2.Visible = True
frmDiagrama1.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
'Envio del P2 y P0 como dirrecciones
txtDir_P2.Move 6120, 150
txtDirDat_P0.Move 5520, 1640
imgArrow1.Move 5760, 1680
imgArrow2.Move 5880, 160
frmDiagrama1.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
'Segunda fase segundo estado
ALE_Latch_color1
Latch_Eprom_QBcolor1
txtDir_P2.Move 9480, 150
txtDirDat_P0.Move 7320, 1640
imgArrow1.Move 7080, 1680
imgArrow2.Move 9240, 160
frmDiagrama1.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
txtDir_P2.Move 9600, 630
txtDirDat_P0.Move 7800, 920
imgArrow1.Move 8040, 960
imgArrow2.Picture = frmImagenes.imgLeftarrow1.Picture
imgArrow2.Move 9360, 660
frmDiagrama1.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
txtDir_P2.Visible = False
txtDirDat_P0.Visible = False
imgArrow1.Visible = False
imgArrow1.Move 8040, 2040
imgArrow2.Visible = False
imgArrow2.Picture = frmImagenes.imgRightarrow1.Picture
imgArrow2.Move 2520, 1200
Label4.ForeColor = QBColor(0)
'Primera fase tercer estado, segunda fase tercer estado,
'primera fase cuarto estado
PSEN_Eprom_color1
P0_Latch_color1
Latch_Eprom_color1
frmDiagrama1.Refresh
Start = Timer: Do While Timer < Start + tiem * 4: Loop
txtDirDat_P0.Move 7800, 2010
txtDirDat_P0.Text = sInstruccionesOperandos(ejec1, 16)
If icolor1 = 1 Then
   Label5.Caption = "Traida de instruccion"
   Label5.Visible = True
   Label5.ForeColor = QBColor(4)
   txtDirDat_P0.Visible = True
   imgArrow1.Picture = frmImagenes.imgLeftarrow1.Picture
   imgArrow1.Visible = True
Else
   Label5.Caption = "Lectura no válida"
   Label5.Visible = True
End If
P0_Eprom_QBcolor1
frmDiagrama1.Refresh
Start = Timer: Do While Timer < Start + 0.7 * tiem: Loop
txtDirDat_P0.Move 5640, 2660
imgArrow1.Move 5880, 2690
frmDiagrama1.Refresh
Start = Timer: Do While Timer < Start + 0.6 * tiem: Loop
txtDirDat_P0.Move 2760, 2180
imgArrow1.Move 2520, 2220
frmDiagrama1.Refresh
Start = Timer: Do While Timer < Start + 0.7 * tiem: Loop
```

```
txtDirDat_P0.Visible = False
imgArrow1.Visible = False
imgArrow1.Picture = frmImagenes.imgRightarrow1.Picture
txtDir_P2.Move 2760, 1190
txtDirDat_P0.Move 2760, 2180
Label4.Visible = False
Label5.Visible = False
frmDiagrama1.Refresh
Next icolor1
'P2_Eprom_QBcolor1
label1.Visible = False
Label2.Visible = False
Label3.Visible = False
frmDiagrama1.Refresh
'Para control de los Timers 0 ^ 1
'Solo en la finalización de cada ciclo maquina
Form2.Temp_Cont_0
Form2.Temp_Cont_1
Form2.Atencion_Interrupciones
End Sub

Sub Byte2_1Ciclo_1()
LecturaP11
'Cada fase tendrá una duración de 2tiem segundo
tiem = Frecuencia_Oscilador
icolor = 16777215  'Line1.BorderColor
icolor2 = 16777215  'Shape1.BackColor
icolor3 = 8421504  'Shape4.BackColor
label1.Caption = saMatrizInstrucciones(ejec1)
Label2.Caption = "#Bytes = " & sInstruccionesOperandos(ejec1, 8)
Label3.Caption = "#C.Maq = " & sInstruccionesOperandos(ejec1, 9)
label1.Visible = True
Label2.Visible = True
Label3.Visible = True
For icolor1 = 1 To 2
txtDir_P2.Text = sInstruccionesOperandos(ejec1, 19 + 2 * (icolor1 - 1))
txtDirDat_P0.Text = sInstruccionesOperandos(ejec1, 20 + 2 * (icolor1 - 1))
'Segunda fase primer estado, primera fase segundo estado,
ALE_Latch_QBcolor1
PSEN_Eprom_QBcolor1
P2_Eprom_QBcolor1
P0_Eprom_color1
frmDiagrama1.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
P2_Eprom_color1
frmDiagrama1.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
P2_Eprom_QBcolor1
P0_Latch_QBcolor1
Label4.Caption = "Direccionamiento a ROM" & " " & "(" & icolor1 & " " & "Byte)"
Label4.Visible = True
Label4.ForeColor = QBColor(4)
txtDir_P2.Visible = True
txtDirDat_P0.Visible = True
imgArrow1.Visible = True
imgArrow2.Visible = True
frmDiagrama1.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
'Envio del P2 y P0 como dirrecciones
txtDir_P2.Move 6120, 150
txtDirDat_P0.Move 5520, 1640
imgArrow1.Move 5760, 1680
imgArrow2.Move 5880, 160
frmDiagrama1.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
'Segunda fase segundo estado
ALE_Latch_color1
PSEN_Eprom_QBcolor1
Latch_Eprom_QBcolor1
txtDir_P2.Move 9480, 150
txtDirDat_P0.Move 7320, 1640
```

```
imgArrow1.Move 7080, 1680
imgArrow2.Move 9240, 160
frmDiagrama1.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
txtDir_P2.Move 9600, 630
txtDirDat_P0.Move 7800, 920
imgArrow1.Move 8040, 960
imgArrow2.Move 9360, 660
imgArrow2.Picture = frmImagenes.imgLeftarrow1.Picture
frmDiagrama1.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
txtDir_P2.Visible = False
txtDirDat_P0.Visible = False
imgArrow1.Visible = False
imgArrow2.Visible = False
imgArrow2.Picture = frmImagenes.imgRightarrow1.Picture
imgArrow2.Move 2520, 1200
Label4.ForeColor = QBColor(0)
'Primera fase tercer estado, segunda fase tercer estado,
'primera fase cuarto estado
PSEN_Eprom_color1
P0_Latch_color1
Latch_Eprom_color1
frmDiagrama1.Refresh
'Para lectura de los interruptores del Puerto P1
If icolor1 = 2 Then
   Start = Timer: Do While Timer < Start + tiem * 2: Loop
   If sInstruccionesOperandos(ejec1, 3) = "144" Or iLecturaP1 = 1 Then
      'Para lectura de todos los bits
      Puerto_P1_QBcolor1
      frmDiagrama1.Refresh
      Start = Timer: Do While Timer < Start + tiem: Loop
      txtPuerto_P1.Text = sValor_InH_P11
      imgArrow_P1.Picture = frmImagenes.imgLeftarrow1.Picture
      txtPuerto_P1.Visible = True
      imgArrow_P1.Visible = True
      frmDiagrama1.Refresh
      Start = Timer: Do While Timer < Start + tiem: Loop
      txtPuerto_P1 = ""
      txtPuerto_P1.Visible = False
      imgArrow_P1.Visible = False
      imgArrow_P1.Picture = frmImagenes.imgRightarrow1.Picture
      Puerto_P1_color1
      For ip1in = 0 To 7
         iMemoriaRAMint(144, ip1in) = saValor_InB_P11(ip1in)
      Next ip1in
      ActualizacionRAM
   ElseIf Mid(sInstruccionesOperandos(ejec1, 3), 1, 2) = "P1" _
         Or Mid(sInstruccionesOperandos(ejec1, 3), 1, 3) = "/P1" Or iLecturaP1 = 2 Then
      'Para de uno de los 4 pines de entrada en  P1
      Puerto_P1_QBcolor1
      frmDiagrama1.Refresh
      Start = Timer: Do While Timer < Start + tiem: Loop
      txtPuerto_P1.ForeColor = QBColor(12)
      If iLecturaP1 = 2 Then
         txtPuerto_P1.Text = Mid(sInstruccionesOperandos(ejec1, 2), 3, 4)
      Else
         txtPuerto_P1.Text = Mid(sInstruccionesOperandos(ejec1, 3), 3, 4)
      End If
      imgArrow_P1.Picture = frmImagenes.imgLeftarrow1.Picture
      txtPuerto_P1.Visible = True
      imgArrow_P1.Visible = True
      frmDiagrama1.Refresh
      Start = Timer: Do While Timer < Start + tiem: Loop
      txtPuerto_P1.ForeColor = QBColor(0)
      txtPuerto_P1 = ""
      txtPuerto_P1.Visible = False
      imgArrow_P1.Visible = False
      imgArrow_P1.Picture = frmImagenes.imgRightarrow1.Picture
      Puerto_P1_color1
      Form2.Puerto_P1RD
```

```
    Else
      Start = Timer: Do While Timer < Start + tiem * 2: Loop
    End If
  Else
    Start = Timer: Do While Timer < Start + tiem * 4: Loop
  End If
  txtDirDat_P0.Move 7800, 2010
  imgArrow1.Picture = frmImagenes.imgLeftarrow1.Picture
  imgArrow1.Move 8040, 2040
  Label5.Caption = "Traida de instruccion" & " " & "(" & icolor1 & " " & "Byte)"
  Label5.Visible = True
  Label5.ForeColor = QBColor(4)
  txtDirDat_P0.Text = sInstruccionesOperandos(ejec1, 16 + icolor1 - 1)
  txtDirDat_P0.Visible = True
  imgArrow1.Visible = True
  P0_Eprom_QBcolor1
  frmDiagrama1.Refresh
  Start = Timer: Do While Timer < Start + 0.7 * tiem: Loop
  txtDirDat_P0.Move 5640, 2660
  imgArrow1.Move 5880, 2690
  frmDiagrama1.Refresh
  Start = Timer: Do While Timer < Start + 0.6 * tiem: Loop
  txtDirDat_P0.Move 2760, 2180
  imgArrow1.Move 2520, 2220
  frmDiagrama1.Refresh
  Start = Timer: Do While Timer < Start + 0.7 * tiem: Loop
  txtDirDat_P0.Visible = False
  imgArrow1.Visible = False
  imgArrow1.Picture = frmImagenes.imgRightarrow1.Picture
  Label4.Visible = False
  Label5.Visible = False
  frmDiagrama1.Refresh
  txtDir_P2.Move 2760, 1190
  txtDirDat_P0.Move 2760, 2180
  Next icolor1
  'Finaliza el único ciclo de maquina
  Form2.Temp_Cont_0
  Form2.Temp_Cont_1
  label1.Visible = False
  Label2.Visible = False
  Label3.Visible = False
  frmDiagrama1.Refresh
  Form2.Atencion_Interrupciones
  'Manejo de salida serial
  If sInstruccionesOperandos(ejec1, 2) = "153" Then
    label1.Caption = saMatrizInstrucciones(ejec1 + 1)
    label1.Visible = True
    lblBitValor.Left = 1005
    lblBitValor.Top = 4110
    For ins = 0 To 9
      TXD_OutSerial_QBcolor1
      Label10.Caption = "00"
      Label10.Visible = True
      Image1.Picture = frmImagenes.imgLeftarrow1.Picture
      Image1.Height = 135
      Image1.Width = 255
      Image1.Left = 810
      Image1.Top = 2010
      Image1.Visible = True
      If ins = 0 Then
        lblBitValor.Caption = "Start/0"
        lblBitValor.Visible = True
      ElseIf ins = 9 Then
        lblBitValor.Caption = "Stop/1"
        lblBitValor.Visible = True
      Else
        lblBitValor.Caption = (ins - 1) & "/" & iMemoriaRAMint(224, 8 - ins)
        lblBitValor.Visible = True
      End If
      frmDiagrama1.Refresh
      For ins1 = 1 To 10
```

```
        Label10.Caption = ins1
        frmDiagrama1.Refresh
        Start = Timer: Do While Timer < Start + 0.2 * tiem: Loop
      Next ins1
      Image1.Visible = False
      Image1.Picture = frmImagenes.imgDownarrow1.Picture
      Image1.Height = 255
      Image1.Width = 135
      Image1.Left = 540
      Image1.Top = 3240
      Image1.Visible = True
      frmDiagrama1.Refresh
      For ins1 = 11 To 20
        Label10.Caption = ins1
        frmDiagrama1.Refresh
        Start = Timer: Do While Timer < Start + 0.2 * tiem: Loop
      Next ins1
      Image1.Visible = False
      Image1.Picture = frmImagenes.imgRightarrow1.Picture
      Image1.Height = 135
      Image1.Width = 255
      Image1.Left = 1965
      Image1.Top = 4140
      Image1.Visible = True
      frmDiagrama1.Refresh
      For ins1 = 21 To 32
        Label10.Caption = ins1
        frmDiagrama1.Refresh
        Start = Timer: Do While Timer < Start + 0.2 * tiem: Loop
      Next ins1
      If ins = 0 Then
        lblBit_Serial(9 - ins).Caption = 0
      ElseIf ins = 9 Then
        lblBit_Serial(9 - ins).Caption = 1
      Else
        lblBit_Serial(9 - ins).Caption = iMemoriaRAMint(224, 8 - ins)
      End If
      Image1.Visible = False
      Label10.Visible = False
      lblBitValor.Visible = False
      TXD_OutSerial_color1
      frmDiagrama1.Refresh
    Next ins
    label1.Visible = False
    iMemoriaRAMint(152, 6) = 1  'Activamos TI
    ActualizacionRAM
  End If
  End Sub

  Sub Byte2_2Ciclo_1()
  LecturaP11
  'Cada fase tendrá una duración de 2tiem segundo
  tiem = Frecuencia_Oscilador
  icolor = 16777215   'Line1.BorderColor
  icolor2 = 16777215  'Shape1.BackColor
  icolor3 = 8421504   'Shape4.BackColor
  label1.Caption = saMatrizInstrucciones(ejec1)
  Label2.Caption = "#Bytes = " & sInstruccionesOperandos(ejec1, 8)
  Label3.Caption = "#C.Maq = " & sInstruccionesOperandos(ejec1, 9)
  label1.Visible = True
  Label2.Visible = True
  Label3.Visible = True
  For icolor1 = 1 To 4
  If icolor1 = 1 Or icolor1 = 2 Then
    txtDir_P2.Text = sInstruccionesOperandos(ejec1, 19 + 2 * (icolor1 - 1))
    txtDirDat_P0.Text = sInstruccionesOperandos(ejec1, 20 + 2 * (icolor1 - 1))
  ElseIf icolor1 = 3 Or icolor1 = 4 Then
    txtDirDat_P0.Text = sInstruccionesOperandos(ejec1, 20 + 2)
  End If
  'Segunda fase primer estado, primera fase segundo estado,
  ALE_Latch_QBcolor1
```

```
PSEN_Eprom_QBcolor1
P2_Eprom_QBcolor1
P0_Eprom_color1
frmDiagrama1.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
P2_Eprom_color1
frmDiagrama1.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
P2_Eprom_QBcolor1
P0_Latch_QBcolor1
If icolor1 = 1 Or icolor1 = 2 Then
   Label4.Caption = "Direccionamiento a ROM" & " " & "(" & icolor1 & " " & "Byte)"
Else
   Label4.Caption = "Direccionamiento a ROM"
End If
Label4.Visible = True
Label4.ForeColor = QBColor(4)

txtDir_P2.Visible = True
txtDirDat_P0.Visible = True
imgArrow1.Visible = True
imgArrow2.Visible = True
frmDiagrama1.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
'Envio del P2 y P0 como dirrecciones
txtDir_P2.Move 6120, 150
txtDirDat_P0.Move 5520, 1640
imgArrow1.Move 5760, 1680
imgArrow2.Move 5880, 160
frmDiagrama1.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
'Segunda fase segundo estado
ALE_Latch_color1
PSEN_Eprom_QBcolor1
Latch_Eprom_QBcolor1
txtDir_P2.Move 9480, 150
txtDirDat_P0.Move 7320, 1640
imgArrow1.Move 7080, 1680
imgArrow2.Move 9240, 160
frmDiagrama1.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
txtDir_P2.Move 9600, 630
txtDirDat_P0.Move 7800, 920
imgArrow1.Move 8040, 960
imgArrow2.Picture = frmImagenes.imgLeftarrow1.Picture
imgArrow2.Move 9360, 660
frmDiagrama1.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
txtDir_P2.Visible = False
txtDirDat_P0.Visible = False
imgArrow1.Visible = False
imgArrow2.Visible = False
imgArrow2.Picture = frmImagenes.imgRightarrow1.Picture
imgArrow2.Move 2520, 1200
Label4.ForeColor = QBColor(0)
frmDiagrama1.Refresh
'Primera fase tercer estado, segunda fase tercer estado,
'primera fase cuarto estado
PSEN_Eprom_color1
P0_Latch_color1
Latch_Eprom_color1
frmDiagrama1.Refresh
If icolor1 = 4 Then
   Start = Timer: Do While Timer < Start + tiem * 2: Loop
   If sInstruccionesOperandos(ejec1, 3) = "144" Or iLecturaP1 = 1 Then
      'Para lectura de todos los bits
      Puerto_P1_QBcolor1
      frmDiagrama1.Refresh
      Start = Timer: Do While Timer < Start + tiem: Loop
      txtPuerto_P1.Text = sValor_InH_P1
      imgArrow_P1.Picture = frmImagenes.imgLeftarrow1.Picture
```

```
        txtPuerto_P1.Visible = True
        imgArrow_P1.Visible = True
        frmDiagrama1.Refresh
        Start = Timer: Do While Timer < Start + tiem: Loop
        txtPuerto_P1 = ""
        txtPuerto_P1.Visible = False
        imgArrow_P1.Visible = False
        imgArrow_P1.Picture = frmImagenes.imgRightarrow1.Picture
        Puerto_P1_color1
        Form2.Puerto_P1RD
    ElseIf Mid(sInstruccionesOperandos(ejec1, 3), 1, 2) = "P1" _
          Or Mid(sInstruccionesOperandos(ejec1, 3), 1, 3) = "/P1" Or iLecturaP1 = 2 Then
        'Para de uno de los 4 pines de entrada en  P1
        Puerto_P1_QBcolor1
        frmDiagrama1.Refresh
        Start = Timer: Do While Timer < Start + tiem: Loop
        txtPuerto_P1.ForeColor = QBColor(12)
        If iLecturaP1 = 2 Then
           txtPuerto_P1.Text = Mid(sInstruccionesOperandos(ejec1, 2), 3, 4)
        Else
           txtPuerto_P1.Text = Mid(sInstruccionesOperandos(ejec1, 3), 3, 4)
        End If
        imgArrow_P1.Picture = frmImagenes.imgLeftarrow1.Picture
        txtPuerto_P1.Visible = True
        imgArrow_P1.Visible = True
        frmDiagrama1.Refresh
        Start = Timer: Do While Timer < Start + tiem: Loop
        txtPuerto_P1.ForeColor = QBColor(0)
        txtPuerto_P1 = ""
        txtPuerto_P1.Visible = False
        imgArrow_P1.Visible = False
        imgArrow_P1.Picture = frmImagenes.imgRightarrow1.Picture
        Puerto_P1_color1
        Form2.Puerto_P1RD
    Else
        Start = Timer: Do While Timer < Start + tiem * 2: Loop
    End If
Else
    Start = Timer: Do While Timer < Start + tiem * 4: Loop
End If
txtDirDat_P0.Move 7800, 2010
If icolor1 = 1 Or icolor1 = 2 Then
    Label5.Caption = "Traida de instruccion" & " " & "(" & icolor1 & " " & "Byte)"
    txtDirDat_P0.Text = sInstruccionesOperandos(ejec1, 16 + icolor1 - 1)
    imgArrow1.Picture = frmImagenes.imgLeftarrow1.Picture
    imgArrow1.Move 8040, 2040
    txtDirDat_P0.Visible = True
    imgArrow1.Visible = True
Else
    Label5.Caption = "Lectura no válida"
End If
Label5.Visible = True
Label5.ForeColor = QBColor(4)
P0_Eprom_QBcolor1
frmDiagrama1.Refresh
Start = Timer: Do While Timer < Start + 0.7 * tiem: Loop
txtDirDat_P0.Move 5640, 2660
imgArrow1.Move 5880, 2690
frmDiagrama1.Refresh
Start = Timer: Do While Timer < Start + 0.6 * tiem: Loop
txtDirDat_P0.Move 2760, 2180
imgArrow1.Move 2520, 2220
frmDiagrama1.Refresh
Start = Timer: Do While Timer < Start + 0.7 * tiem: Loop
txtDirDat_P0.Visible = False
imgArrow1.Visible = False
imgArrow1.Picture = frmImagenes.imgRightarrow1.Picture
Label4.Visible = False
Label5.Visible = False
frmDiagrama1.Refresh
txtDir_P2.Move 2760, 1190
```

```
txtDirDat_P0.Move 2760, 2180
If icolor1 = 2 Or icolor1 = 4 Then
   Form2.Temp_Cont_0
   Form2.Temp_Cont_1
End If
Next icolor1
label1.Visible = False
Label2.Visible = False
Label3.Visible = False
frmDiagrama1.Refresh
Form2.Atencion_Interrupciones
End Sub

Sub Byte3_2Ciclo_1()
LecturaP11
'Cada fase tendrá una duración de 2tiem segundo
tiem = Frecuencia_Oscilador
icolor = 16777215   'Line1.BorderColor
icolor2 = 16777215  'Shape1.BackColor
icolor3 = 8421504   'Shape4.BackColor
label1.Caption = saMatrizInstrucciones(ejec1)
Label2.Caption = "#Bytes = " & sInstruccionesOperandos(ejec1, 8)
Label3.Caption = "#C.Maq = " & sInstruccionesOperandos(ejec1, 9)
label1.Visible = True
Label2.Visible = True
Label3.Visible = True
For icolor1 = 1 To 4
If icolor1 = 1 Or icolor1 = 2 Or icolor1 = 3 Then
   txtDir_P2.Text = sInstruccionesOperandos(ejec1, 19 + 2 * (icolor1 - 1))
   txtDirDat_P0.Text = sInstruccionesOperandos(ejec1, 20 + 2 * (icolor1 - 1))
ElseIf icolor1 = 4 Then
   txtDirDat_P0.Text = sInstruccionesOperandos(ejec1, 20 + 4)
End If
'Segunda fase primer estado, primera fase segundo estado,
ALE_Latch_QBcolor1
PSEN_Eprom_QBcolor1
P2_Eprom_QBcolor1
P0_Eprom_color1
frmDiagrama1.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
P2_Eprom_color1
frmDiagrama1.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
P2_Eprom_QBcolor1
P0_Latch_QBcolor1
If icolor1 = 1 Or icolor1 = 2 Or icolor1 = 3 Then
   Label4.Caption = "Direccionamiento a ROM" & " " & "(" & icolor1 & " " & "Byte)"
Else
   Label4.Caption = "Direccionamiento a ROM"
End If
Label4.Visible = True
Label4.ForeColor = QBColor(4)
txtDir_P2.Visible = True
txtDirDat_P0.Visible = True
imgArrow1.Visible = True
imgArrow2.Visible = True
frmDiagrama1.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
'Envio del P2 y P0 como dirrecciones
txtDir_P2.Move 6120, 150
txtDirDat_P0.Move 5520, 1640
imgArrow1.Move 5760, 1680
imgArrow2.Move 5880, 160
frmDiagrama1.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
'Segunda fase segundo estado
ALE_Latch_color1
PSEN_Eprom_QBcolor1
Latch_Eprom_QBcolor1
txtDir_P2.Move 9480, 150
txtDirDat_P0.Move 7320, 1640
```

```
imgArrow1.Move 7080, 1680
imgArrow2.Move 9240, 160
frmDiagrama1.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
txtDir_P2.Move 9600, 630
txtDirDat_P0.Move 7800, 920
imgArrow1.Move 8040, 960
imgArrow2.Picture = frmImagenes.imgLeftarrow1.Picture
imgArrow2.Move 9360, 660
frmDiagrama1.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
txtDir_P2.Visible = False
txtDirDat_P0.Visible = False
imgArrow1.Visible = False
imgArrow2.Visible = False
imgArrow2.Picture = frmImagenes.imgRightarrow1.Picture
imgArrow2.Move 2520, 1200
Label4.ForeColor = QBColor(0)
frmDiagrama1.Refresh
frmDiagrama1.Refresh
'Primera fase tercer estado, segunda fase tercer estado,
'primera fase cuarto estado
ALE_Latch_color1
PSEN_Eprom_color1
P0_Latch_color1
Latch_Eprom_color1
frmDiagrama1.Refresh
If icolor1 = 4 Then
   Start = Timer: Do While Timer < Start + tiem * 2: Loop
   If sInstruccionesOperandos(ejec1, 3) = "144" Or iLecturaP1 = 1 Then
      'Para lectura de todos los bits
      Puerto_P1_QBcolor1
      frmDiagrama1.Refresh
      Start = Timer: Do While Timer < Start + tiem: Loop
      txtPuerto_P1.Text = sValor_InH_P1
      imgArrow_P1.Picture = frmImagenes.imgLeftarrow1.Picture
      txtPuerto_P1.Visible = True
      imgArrow_P1.Visible = True
      frmDiagrama1.Refresh
      Start = Timer: Do While Timer < Start + tiem: Loop
      txtPuerto_P1 = ""
      txtPuerto_P1.Visible = False
      imgArrow_P1.Visible = False
      imgArrow_P1.Picture = frmImagenes.imgRightarrow1.Picture
      Puerto_P1_color1
      Form2.Puerto_P1RD
   ElseIf Mid(sInstruccionesOperandos(ejec1, 3), 1, 2) = "P1" _
         Or Mid(sInstruccionesOperandos(ejec1, 3), 1, 3) = "/P1" Or iLecturaP1 = 2 Then
      'Para de uno de los 4 pines de entrada en  P1
      Puerto_P1_QBcolor1
      frmDiagrama1.Refresh
      Start = Timer: Do While Timer < Start + tiem: Loop
      txtPuerto_P1.ForeColor = QBColor(12)
      If iLecturaP1 = 2 Then
         txtPuerto_P1.Text = Mid(sInstruccionesOperandos(ejec1, 2), 3, 4)
      Else
         txtPuerto_P1.Text = Mid(sInstruccionesOperandos(ejec1, 3), 3, 4)
      End If
      imgArrow_P1.Picture = frmImagenes.imgLeftarrow1.Picture
      txtPuerto_P1.Visible = True
      imgArrow_P1.Visible = True
      frmDiagrama1.Refresh
      Start = Timer: Do While Timer < Start + tiem: Loop
      txtPuerto_P1.ForeColor = QBColor(0)
      txtPuerto_P1 = ""
      txtPuerto_P1.Visible = False
      imgArrow_P1.Visible = False
      imgArrow_P1.Picture = frmImagenes.imgRightarrow1.Picture
      Puerto_P1_color1
      Form2.Puerto_P1RD
   Else
```

```
        Start = Timer: Do While Timer < Start + tiem * 2: Loop
    End If
  Else
    Start = Timer: Do While Timer < Start + tiem * 4: Loop
  End If
  txtDirDat_P0.Move 7800, 2010
  If icolor1 = 1 Or icolor1 = 2 Or icolor1 = 3 Then
    Label5.Caption = "Traida de instruccion" & " " & "(" & icolor1 & " " & "Byte)"
    txtDirDat_P0.Text = sInstruccionesOperandos(ejec1, 16 + icolor1 - 1)
    imgArrow1.Picture = frmImagenes.imgLeftarrow1.Picture
    imgArrow1.Move 8040, 2040
    txtDirDat_P0.Visible = True
    imgArrow1.Visible = True
  Else
    Label5.Caption = "Lectura no válida"
  End If
  Label5.Visible = True
  Label5.ForeColor = QBColor(4)
  P0_Eprom_QBcolor1
  frmDiagrama1.Refresh
  Start = Timer: Do While Timer < Start + 0.7 * tiem: Loop
  txtDirDat_P0.Move 5640, 2660
  imgArrow1.Move 5880, 2690
  frmDiagrama1.Refresh
  Start = Timer: Do While Timer < Start + 0.6 * tiem: Loop
  txtDirDat_P0.Move 2760, 2180
  imgArrow1.Move 2520, 2220
  frmDiagrama1.Refresh
  Start = Timer: Do While Timer < Start + 0.7 * tiem: Loop
  txtDirDat_P0.Visible = False
  imgArrow1.Visible = False
  mgArrow1.Picture = frmImagenes.imgRightarrow1.Picture
  Label4.Visible = False
  Label5.Visible = False
  frmDiagrama1.Refresh
  txtDir_P2.Move 2760, 1190
  txtDirDat_P0.Move 2760, 2180
  If icolor1 = 2 Or icolor1 = 4 Then
    Form2.Temp_Cont_0
    Form2.Temp_Cont_1
  End If
  Next icolor1
  label1.Visible = False
  Label2.Visible = False
  Label3.Visible = False
  frmDiagrama1.Refresh
  Form2.Atencion_Interrupciones
  'Manejo de entrada serial
  If sInstruccionesOperandos(ejec1, 3) = "#70H" Then
    label1.Caption = saMatrizInstrucciones(ejec1 + 1)
    label1.Visible = True
    lblBitValor.Left = 1005
    lblBitValor.Top = 4830
    For ins = 0 To 9
      RXD_InSerial_QBcolor1
      Label10.Caption = "00"
      Label10.Visible = True
      Image1.Picture = frmImagenes.imgLeftarrow1.Picture
      Image1.Height = 135
      Image1.Width = 255
      Image1.Left = 1965
      Image1.Top = 4860
      Image1.Visible = True
      If ins = 0 Then
        lblBitValor.Caption = "Start/0"
        lblBitValor.Visible = True
      ElseIf ins = 9 Then
        lblBitValor.Caption = "Stop/1"
        lblBitValor.Visible = True
      Else
        lblBitValor.Caption = (ins - 1) & "/" & lblBit_Serial(10 + ins).Caption
```

```
        lblBitValor.Visible = True
      End If
      frmDiagrama1.Refresh
      For ins1 = 1 To 5
        Label10.Caption = ins1
        frmDiagrama1.Refresh
        Start = Timer: Do While Timer < Start + 0.2 * tiem: Loop
      Next ins1
      Image1.Visible = False
      Image1.Picture = frmImagenes.imgUparrow1.Picture
      Image1.Height = 255
      Image1.Width = 135
      Image1.Left = 420
      Image1.Top = 3240
      Image1.Visible = True
      frmDiagrama1.Refresh
      For ins1 = 6 To 11
        Label10.Caption = ins1
        frmDiagrama1.Refresh
        Start = Timer: Do While Timer < Start + 0.2 * tiem: Loop
      Next ins1
      Image1.Visible = False
      Image1.Picture = frmImagenes.imgRightarrow1.Picture
      Image1.Height = 135
      Image1.Width = 255
      Image1.Left = 810
      Image1.Top = 1920
      Image1.Visible = True
      frmDiagrama1.Refresh
      For ins1 = 12 To 16
        Label10.Caption = ins1
        frmDiagrama1.Refresh
        Start = Timer: Do While Timer < Start + 0.2 * tiem: Loop
      Next ins1
      Image1.Visible = False
      Label10.Visible = False
      lblBitValor.Visible = False
      RXD_InSerial_color1
      frmDiagrama1.Refresh
    Next ins
    label1.Visible = False
    iMemoriaRAMintD(153) = iValor_In_InS
    ActualizacionRAMD_BH
  End If
  If sInstruccionesOperandos(ejec1, 0) = "RX" Then
    iMemoriaRAMint(152, 7) = 1 'Activamos RI
    ActualizacionRAM
  End If
  End Sub

  Sub Latch_Eprom_color1()
  frmDiagrama1.Shape4.BackColor = icolor3
  frmDiagrama1.Shape5.BackColor = icolor3
  frmDiagrama1.Shape4.BorderColor = icolor3
  frmDiagrama1.Shape5.BorderColor = icolor3
  frmDiagrama1.Line32.BorderColor = icolor3
  frmDiagrama1.Line33.BorderColor = icolor3
  frmDiagrama1.Line34.BorderColor = icolor3
  End Sub

  Sub Latch_Eprom_QBcolor1()
  frmDiagrama1.Shape4.BackColor = QBColor(3)
  frmDiagrama1.Shape5.BackColor = QBColor(3)
  frmDiagrama1.Shape4.BorderColor = QBColor(3)
  frmDiagrama1.Shape5.BorderColor = QBColor(3)
  frmDiagrama1.Line32.BorderColor = QBColor(3)
  frmDiagrama1.Line33.BorderColor = QBColor(3)
  frmDiagrama1.Line34.BorderColor = QBColor(3)
  End Sub

  Sub Latch_Ram_color1()
```

```
frmDiagrama1.Shape4.BackColor = icolor3
frmDiagrama1.Shape8.BackColor = icolor3
frmDiagrama1.Shape4.BorderColor = icolor3
frmDiagrama1.Shape8.BorderColor = icolor3
frmDiagrama1.Line32.BorderColor = icolor3
frmDiagrama1.Line42.BorderColor = icolor3
frmDiagrama1.Line43.BorderColor = icolor3
End Sub

Sub Latch_Ram_QBcolor1()
frmDiagrama1.Shape4.BackColor = QBColor(3)
frmDiagrama1.Shape8.BackColor = QBColor(3)
frmDiagrama1.Shape4.BorderColor = QBColor(3)
frmDiagrama1.Shape8.BorderColor = QBColor(3)
frmDiagrama1.Line32.BorderColor = QBColor(3)
frmDiagrama1.Line42.BorderColor = QBColor(3)
frmDiagrama1.Line43.BorderColor = QBColor(3)
End Sub

Sub LecturaP11()
If sInstruccionesOperandos(ejec1, 1) = "PUSH" _
  And sInstruccionesOperandos(ejec1, 2) = "P1" Then
   iLecturaP1 = 1 'Si es byte
ElseIf (sInstruccionesOperandos(ejec1, 1) = "JB" _
     Or sInstruccionesOperandos(ejec1, 1) = "JNB") _
     And (Mid(sInstruccionesOperandos(ejec1, 2), 1, 2) = "P1" _
     Or Mid(sInstruccionesOperandos(ejec1, 2), 1, 3) = "/P1") Then
      iLecturaP1 = 2 'Si es bit
Else
   iLecturaP1 = 5  'Un valor cualquiera diferente de 1 y 2
End If
End Sub

Sub P0_Eprom_color1()
frmDiagrama1.Shape7.BackColor = icolor2
frmDiagrama1.Shape2.BackColor = icolor2
frmDiagrama1.Shape7.BorderColor = icolor2
frmDiagrama1.Shape2.BorderColor = icolor2
frmDiagrama1.Line39.BorderColor = icolor2
frmDiagrama1.Line38.BorderColor = icolor2
frmDiagrama1.Line37.BorderColor = icolor2
frmDiagrama1.Line36.BorderColor = icolor2
frmDiagrama1.Line35.BorderColor = icolor2
frmDiagrama1.Line29.BorderColor = icolor2
End Sub

Sub P0_Eprom_QBcolor1()
frmDiagrama1.Shape7.BackColor = QBColor(6)
frmDiagrama1.Shape2.BackColor = QBColor(6)
frmDiagrama1.Shape7.BorderColor = QBColor(6)
frmDiagrama1.Shape2.BorderColor = QBColor(6)
frmDiagrama1.Line39.BorderColor = QBColor(6)
frmDiagrama1.Line38.BorderColor = QBColor(6)
frmDiagrama1.Line37.BorderColor = QBColor(6)
frmDiagrama1.Line36.BorderColor = QBColor(6)
frmDiagrama1.Line35.BorderColor = QBColor(6)
frmDiagrama1.Line29.BorderColor = QBColor(6)
End Sub

Sub P0_Latch_color1()
frmDiagrama1.Shape2.BackColor = icolor2
frmDiagrama1.Shape3.BackColor = icolor2
frmDiagrama1.Shape2.BorderColor = icolor2
frmDiagrama1.Shape3.BorderColor = icolor2
frmDiagrama1.Line29.BorderColor = icolor2
frmDiagrama1.Line30.BorderColor = icolor2
frmDiagrama1.Line31.BorderColor = icolor2
End Sub

Sub P0_Latch_QBcolor1()
frmDiagrama1.Shape2.BackColor = QBColor(3)
```

```
frmDiagrama1.Shape3.BackColor = QBColor(3)
frmDiagrama1.Shape2.BorderColor = QBColor(3)
frmDiagrama1.Shape3.BorderColor = QBColor(3)
frmDiagrama1.Line29.BorderColor = QBColor(3)
frmDiagrama1.Line30.BorderColor = QBColor(3)
frmDiagrama1.Line31.BorderColor = QBColor(3)
End Sub

Sub P0_Ram_color1()
frmDiagrama1.Shape2.BackColor = icolor2
frmDiagrama1.Shape10.BackColor = icolor2
frmDiagrama1.Shape2.BorderColor = icolor2
frmDiagrama1.Shape10.BorderColor = icolor2
frmDiagrama1.Line29.BorderColor = icolor2
frmDiagrama1.Line35.BorderColor = icolor2
frmDiagrama1.Line36.BorderColor = icolor2
frmDiagrama1.Line40.BorderColor = icolor2
frmDiagrama1.Line41.BorderColor = icolor2
End Sub

Sub P0_Ram_QBcolor1()
frmDiagrama1.Shape2.BackColor = QBColor(6)
frmDiagrama1.Shape10.BackColor = QBColor(6)
frmDiagrama1.Shape2.BorderColor = QBColor(6)
frmDiagrama1.Shape10.BorderColor = QBColor(6)
frmDiagrama1.Line29.BorderColor = QBColor(6)
frmDiagrama1.Line35.BorderColor = QBColor(6)
frmDiagrama1.Line36.BorderColor = QBColor(6)
frmDiagrama1.Line40.BorderColor = QBColor(6)
frmDiagrama1.Line41.BorderColor = QBColor(6)
End Sub

Sub P2_Eprom_color1()
frmDiagrama1.Shape1.BackColor = icolor2
frmDiagrama1.Shape6.BackColor = icolor2
frmDiagrama1.Shape1.BorderColor = icolor2
frmDiagrama1.Shape6.BorderColor = icolor2
frmDiagrama1.Line22.BorderColor = icolor2
frmDiagrama1.Line23.BorderColor = icolor2
frmDiagrama1.Line24.BorderColor = icolor2
frmDiagrama1.Line25.BorderColor = icolor2
frmDiagrama1.Line26.BorderColor = icolor2
End Sub

Sub P2_Eprom_QBcolor1()
frmDiagrama1.Shape1.BackColor = QBColor(3)
frmDiagrama1.Shape6.BackColor = QBColor(3)
frmDiagrama1.Shape1.BorderColor = QBColor(3)
frmDiagrama1.Shape6.BorderColor = QBColor(3)
frmDiagrama1.Line22.BorderColor = QBColor(3)
frmDiagrama1.Line23.BorderColor = QBColor(3)
frmDiagrama1.Line24.BorderColor = QBColor(3)
frmDiagrama1.Line25.BorderColor = QBColor(3)
frmDiagrama1.Line26.BorderColor = QBColor(3)
End Sub

Sub P2_Ram_color1()
frmDiagrama1.Shape1.BackColor = icolor2
frmDiagrama1.Shape9.BackColor = icolor2
frmDiagrama1.Shape1.BorderColor = icolor2
frmDiagrama1.Shape9.BorderColor = icolor2
frmDiagrama1.Line22.BorderColor = icolor2
frmDiagrama1.Line23.BorderColor = icolor2
frmDiagrama1.Line24.BorderColor = icolor2
frmDiagrama1.Line25.BorderColor = icolor2
frmDiagrama1.Line27.BorderColor = icolor2
frmDiagrama1.Line28.BorderColor = icolor2
End Sub

Sub P2_Ram_QBcolor1()
frmDiagrama1.Shape1.BackColor = QBColor(3)
```

```
frmDiagrama1.Shape9.BackColor = QBColor(3)
frmDiagrama1.Shape1.BorderColor = QBColor(3)
frmDiagrama1.Shape9.BorderColor = QBColor(3)
frmDiagrama1.Line22.BorderColor = QBColor(3)
frmDiagrama1.Line23.BorderColor = QBColor(3)
frmDiagrama1.Line24.BorderColor = QBColor(3)
frmDiagrama1.Line25.BorderColor = QBColor(3)
frmDiagrama1.Line27.BorderColor = QBColor(3)
frmDiagrama1.Line28.BorderColor = QBColor(3)
End Sub

Sub PSEN_Eprom_color1()
frmDiagrama1.Line4.BorderColor = icolor
frmDiagrama1.Line5.BorderColor = icolor
frmDiagrama1.Line6.BorderColor = icolor
frmDiagrama1.Line7.BorderColor = icolor
End Sub

Sub PSEN_Eprom_QBcolor1()
frmDiagrama1.Line4.BorderColor = QBColor(12)
frmDiagrama1.Line5.BorderColor = QBColor(12)
frmDiagrama1.Line6.BorderColor = QBColor(12)
frmDiagrama1.Line7.BorderColor = QBColor(12)
End Sub

Sub Puerto_P1_color1()
frmDiagrama1.Line44.BorderColor = icolor
frmDiagrama1.Shape11.BorderColor = icolor
frmDiagrama1.Shape11.BackColor = icolor
End Sub

Sub Puerto_P1_QBcolor1()
frmDiagrama1.Line44.BorderColor = QBColor(6)
frmDiagrama1.Shape11.BorderColor = QBColor(6)
frmDiagrama1.Shape11.BackColor = QBColor(6)
End Sub

Sub RXD_InSerial_color1()
frmDiagrama1.Line48.BorderColor = icolor
frmDiagrama1.Line49.BorderColor = icolor
frmDiagrama1.Line50.BorderColor = icolor
End Sub

Sub RXD_InSerial_QBcolor1()
frmDiagrama1.Line48.BorderColor = QBColor(12)
frmDiagrama1.Line49.BorderColor = QBColor(12)
frmDiagrama1.Line50.BorderColor = QBColor(12)
End Sub

Sub TXD_OutSerial_color1()
frmDiagrama1.Line45.BorderColor = icolor
frmDiagrama1.Line46.BorderColor = icolor
frmDiagrama1.Line47.BorderColor = icolor
End Sub

Sub TXD_OutSerial_QBcolor1()
frmDiagrama1.Line45.BorderColor = QBColor(12)
frmDiagrama1.Line46.BorderColor = QBColor(12)
frmDiagrama1.Line47.BorderColor = QBColor(12)
End Sub

Private Sub Form_Load()
iValor_In_P11 = 255
For ip11 = 0 To 7
   saValor_InB_P11(ip11) = 1
Next ip11
sValor_InH_P11 = "FF"
iValor_In_InS = 0
For ip11 = 0 To 7
   saValor_InB_InS(ip11) = 0
Next ip11
```

```
sValor_InH_InS = "00"
End Sub

Private Sub img2KRam_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)
If Button = 2 Then  'Sólo para el boton derecho
   PopupMenu Form2.mnuMemoriasRAM, vbPopupMenuLeftAlign
End If
End Sub

Private Sub img2KRom_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)
If Button = 2 Then  'Sólo para el boton derecho
   PopupMenu Form2.mnuMemoriasROM, vbPopupMenuLeftAlign
End If
End Sub

Private Sub imgLatch_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)
If Button = 2 Then  'Sólo para el boton derecho
   PopupMenu Form2.mnuLatchz, vbPopupMenuLeftAlign
End If
End Sub

Private Sub imgMicro_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)
If Button = 2 Then  'Sólo para el boton derecho
   PopupMenu Form2.mnuMicro, vbPopupMenuLeftAlign
End If
End Sub

Private Sub imgPuerto_P1_Click(Index As Integer)
   If imgPuerto_P1(Index).Tag = "1" Then
      imgPuerto_P1(Index).Picture = frmImagenes.imgP1_01.Picture
      imgPuerto_P1(Index).Tag = "0"
   Else
      imgPuerto_P1(Index).Picture = frmImagenes.imgP1_11.Picture
      imgPuerto_P1(Index).Tag = "1"
   End If
   For ip1in = 0 To 7
      saValor_InB_P11(ip1in) = Val(imgPuerto_P1(7 - ip1in).Tag)
   Next ip1in
   For ip1in = 0 To 7
      iValorB(ip1in) = saValor_InB_P11(ip1in)
   Next ip1in
   ConversionB_D
   iValor_In_P11 = iValorD
   iX = iValorD
   ConversionD_H
   sValor_InH_P11 = sValorH
End Sub

Private Sub lblBit_Serial_Click(Index As Integer)
If Index >= 11 And Index <= 18 Then
   If lblBit_Serial(Index).Caption = "1" Then
      lblBit_Serial(Index).Caption = "0"
   Else
      lblBit_Serial(Index).Caption = "1"
   End If
   For ip1in = 11 To 18
      saValor_InB_InS(18 - ip1in) = Val(lblBit_Serial(ip1in).Caption)
   Next ip1in
   For ip1in = 0 To 7
      iValorB(ip1in) = saValor_InB_InS(ip1in)
   Next ip1in
   ConversionB_D
   iValor_In_InS = iValorD
   iX = iValorD
   ConversionD_H
   sValor_InH_InS = sValorH
End If
End Sub
```

frmDiagrama2

```
'Declaración de variables
Dim icolor As Long
Dim icolor2 As Long
Dim icolor3 As Long
Dim iRamBufferLatch2 As Byte

Sub ALE_Latch_color1()
Line1.BorderColor = icolor
Line2.BorderColor = icolor
Line3.BorderColor = icolor
End Sub

Sub ALE_Latch_QBcolor1()
Line1.BorderColor = QBColor(12)
Line2.BorderColor = QBColor(12)
Line3.BorderColor = QBColor(12)
End Sub

Sub Byte1_1Ciclo_1()
EscrituraP1
'Cada fase tendrá una duración de 2*tiem segundos
tiem = Frecuencia_Oscilador
icolor = 16777215  'Line1.BorderColor
icolor2 = 16777215  'Shape1.BackColor
icolor3 = 8421504  'Shape4.BackColor
For icolor1 = 1 To 2
txtDir_P2.Text = sInstruccionesOperandos(ejec1, 19)
txtDirDat_P0.Text = sInstruccionesOperandos(ejec1, 20)
'Segunda fase primer estado, primera fase segundo estado,
label1.Caption = saMatrizInstrucciones(ejec1)
Label2.Caption = "#Bytes = " & sInstruccionesOperandos(ejec1, 8)
Label3.Caption = "#C.Maq = " & sInstruccionesOperandos(ejec1, 9)
label1.Visible = True
Label2.Visible = True
Label3.Visible = True
ALE_Latch_QBcolor1
PSEN_Eprom_QBcolor1
P2_Eprom_QBcolor1
P0_Eprom_color1
'Para Puerto P1 en WR
   If icolor1 = 1 And ejec1 > 0 And iEscrituraP1 Then
      If sInstruccionesOperandos(ejec1 - 1, 2) = "144" Or Mid(sInstruccionesOperandos(ejec1 - 1, 2), 1,
                                                                                          2) = "P1" Then
         Puerto_P1_QBcolor1
      End If
   End If
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
P2_Eprom_color1
'Para Puerto P1 en WR
   If icolor1 = 1 And ejec1 > 0 And iEscrituraP1 Then
      If sInstruccionesOperandos(ejec1 - 1, 2) = "144" Then
         txtPuerto_P1.Text = sMemoriaRAMintH(144)
         txtPuerto_P1.Visible = True
         imgArrow_P1.Visible = True
      ElseIf Mid(sInstruccionesOperandos(ejec1 - 1, 2), 1, 2) = "P1" Then
         txtPuerto_P1.ForeColor = QBColor(12)
         txtPuerto_P1.Text = Mid(sInstruccionesOperandos(ejec1 - 1, 2), 3, 4)
         txtPuerto_P1.Visible = True
         imgArrow_P1.Visible = True
      End If
   End If
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
   If icolor1 = 1 And ejec1 > 0 And iEscrituraP1 Then
      If sInstruccionesOperandos(ejec1 - 1, 2) = "144" Or Mid(sInstruccionesOperandos(ejec1 - 1, 2), 1,
                                                                                          2) = "P1" Then
         txtPuerto_P1 = ""
         txtPuerto_P1.Visible = False
```

```
            imgArrow_P1.Visible = False
            txtPuerto_P1.ForeColor = QBColor(0)
            Form2.Puerto_P1WR
            Puerto_P1_color1
        End If
    End If
P2_Eprom_QBcolor1
P0_Latch_QBcolor1
'El direccionamiento realmente comienza en este estado
Label4.Caption = "Direccionamiento a ROM"
Label4.Visible = True
Label4.ForeColor = QBColor(4)
txtDir_P2.Visible = True
txtDirDat_P0.Visible = True
imgArrow1.Visible = True
imgArrow2.Visible = True
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
'Envio del P2 y P0 como dirrecciones
txtDir_P2.Move 6120, 150
txtDirDat_P0.Move 5520, 1640
imgArrow1.Move 5760, 1680
imgArrow2.Move 5880, 160
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
'Segunda fase segundo estado
ALE_Latch_color1
Latch_Eprom_QBcolor1
txtDir_P2.Move 9480, 150
txtDirDat_P0.Move 7320, 1640
imgArrow1.Move 7080, 1680
imgArrow2.Move 9240, 160
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
txtDir_P2.Move 9600, 630
txtDirDat_P0.Move 7800, 920
imgArrow1.Move 8040, 960
imgArrow2.Picture = frmImagenes.imgLeftarrow1.Picture
imgArrow2.Move 9360, 660
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
txtDir_P2.Visible = False
txtDirDat_P0.Visible = False
imgArrow1.Visible = False
imgArrow1.Move 8040, 2040
imgArrow2.Visible = False
imgArrow2.Picture = frmImagenes.imgRightarrow1.Picture
imgArrow2.Move 2520, 1200
Label4.ForeColor = QBColor(0)
'Primera fase tercer estado, segunda fase tercer estado,
'primera fase cuarto estado
PSEN_Eprom_color1
P0_Latch_color1
Latch_Eprom_color1
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem * 4: Loop
txtDirDat_P0.Move 7800, 2010
txtDirDat_P0.Text = sInstruccionesOperandos(ejec1, 16)
If icolor1 = 1 Then
    Label5.Caption = "Traida de instruccion"
    Label5.Visible = True
    Label5.ForeColor = QBColor(4)
    txtDirDat_P0.Visible = True
    imgArrow1.Picture = frmImagenes.imgLeftarrow1.Picture
    imgArrow1.Visible = True
Else
    Label5.Caption = "Lectura no válida"
    Label5.Visible = True
End If
P0_Eprom_QBcolor1
frmDiagrama2.Refresh
```

```
Start = Timer: Do While Timer < Start + 0.7 * tiem: Loop
txtDirDat_P0.Move 5640, 2660
imgArrow1.Move 5880, 2690
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 0.6 * tiem: Loop
txtDirDat_P0.Move 2760, 2180
imgArrow1.Move 2520, 2220
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 0.7 * tiem: Loop
txtDirDat_P0.Visible = False
imgArrow1.Visible = False
imgArrow1.Picture = frmImagenes.imgRightarrow1.Picture
txtDir_P2.Move 2760, 1190
txtDirDat_P0.Move 2760, 2180
Label4.Visible = False
Label5.Visible = False
frmDiagrama2.Refresh
Next icolor1
label1.Visible = False
Label2.Visible = False
Label3.Visible = False
frmDiagrama2.Refresh
'Para control de los Timers 0 ^ 1
'Solo en la finalización de cada ciclo maquina
Form2.Temp_Cont_0
Form2.Temp_Cont_1
Form2.Atencion_Interrupciones
End Sub

Sub Byte1_2Ciclo_1()
EscrituraP1
'Cada fase tendrá una duración de 2tiem segundo
tiem = Frecuencia_Oscilador
icolor = 16777215  'Line1.BorderColor
icolor2 = 16777215  'Shape1.BackColor
icolor3 = 8421504  'Shape4.BackColor
label1.Caption = saMatrizInstrucciones(ejec1)
Label2.Caption = "#Bytes = " & sInstruccionesOperandos(ejec1, 8)
Label3.Caption = "#C.Maq = " & sInstruccionesOperandos(ejec1, 9)
label1.Visible = True
Label2.Visible = True
Label3.Visible = True
For icolor1 = 1 To 4
txtDir_P2.Text = sInstruccionesOperandos(ejec1, 19)
txtDirDat_P0.Text = sInstruccionesOperandos(ejec1, 20)
'Segunda fase primer estado, primera fase segundo estado,
ALE_Latch_QBcolor1
PSEN_Eprom_QBcolor1
P2_Eprom_QBcolor1
P0_Eprom_color1
'Para Puerto P1 en WR
  If icolor1 = 1 And ejec1 > 0 And iEscrituraP1 Then
    If sInstruccionesOperandos(ejec1 - 1, 2) = "144" Or Mid(sInstruccionesOperandos(ejec1 - 1, 2), 1,
                                                          2) = "P1" Then
      Puerto_P1_QBcolor1
    End If
  End If
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
P2_Eprom_color1
'Para Puerto P1 en WR
  If icolor1 = 1 And ejec1 > 0 And iEscrituraP1 Then
    If sInstruccionesOperandos(ejec1 - 1, 2) = "144" Then
      txtPuerto_P1.Text = sMemoriaRAMintH(144)
      txtPuerto_P1.Visible = True
      imgArrow_P1.Visible = True
    ElseIf Mid(sInstruccionesOperandos(ejec1 - 1, 2), 1, 2) = "P1" Then
      txtPuerto_P1.ForeColor = QBColor(12)
      txtPuerto_P1.Text = Mid(sInstruccionesOperandos(ejec1 - 1, 2), 3, 4)
      txtPuerto_P1.Visible = True
      imgArrow_P1.Visible = True
```

```
        End If
      End If
    frmDiagrama2.Refresh
    Start = Timer: Do While Timer < Start + tiem: Loop
    'Para Puerto P1 en WR
      If icolor1 = 1 And ejec1 > 0 And iEscrituraP1 Then
        If sInstruccionesOperandos(ejec1 - 1, 2) = "144" Or Mid(sInstruccionesOperandos(ejec1 - 1, 2), 1,
                                                                2) = "P1" Then
          txtPuerto_P1 = ""
          txtPuerto_P1.Visible = False
          imgArrow_P1.Visible = False
          txtPuerto_P1.ForeColor = QBColor(0)
          Form2.Puerto_P1WR
          Puerto_P1_color1
        End If
      End If
    P2_Eprom_QBcolor1
    P0_Latch_QBcolor1
    Label4.Caption = "Direccionamiento a ROM"
    Label4.Visible = True
    Label4.ForeColor = QBColor(4)
    txtDir_P2.Visible = True
    txtDirDat_P0.Visible = True
    imgArrow1.Visible = True
    imgArrow2.Visible = True
    frmDiagrama2.Refresh
    Start = Timer: Do While Timer < Start + tiem: Loop
    'Envio del P2 y P0 como dirrecciones
    txtDir_P2.Move 6120, 150
    txtDirDat_P0.Move 5520, 1640
    imgArrow1.Move 5760, 1680
    imgArrow2.Move 5880, 160
    frmDiagrama2.Refresh
    Start = Timer: Do While Timer < Start + tiem: Loop
    'Segunda fase segundo estado
    ALE_Latch_color1
    Latch_Eprom_QBcolor1
    txtDir_P2.Move 9480, 150
    txtDirDat_P0.Move 7320, 1640
    imgArrow1.Move 7080, 1680
    imgArrow2.Move 9240, 160
    frmDiagrama2.Refresh
    Start = Timer: Do While Timer < Start + tiem: Loop
    txtDir_P2.Move 9600, 630
    txtDirDat_P0.Move 7800, 920
    imgArrow1.Move 8040, 960
    imgArrow2.Picture = frmImagenes.imgLeftarrow1.Picture
    imgArrow2.Move 9360, 660
    frmDiagrama2.Refresh
    Start = Timer: Do While Timer < Start + tiem: Loop
    txtDir_P2.Visible = False
    txtDirDat_P0.Visible = False
    imgArrow1.Visible = False
    imgArrow2.Picture = frmImagenes.imgRightarrow1.Picture
    imgArrow2.Visible = False
    imgArrow2.Move 2520, 1200
    Label4.ForeColor = QBColor(0)
    frmDiagrama2.Refresh
    'Primera fase tercer estado, segunda fase tercer estado,
    'primera fase cuarto estado
    PSEN_Eprom_color1
    P0_Latch_color1
    Latch_Eprom_color1
    frmDiagrama2.Refresh
    Start = Timer: Do While Timer < Start + 4 * tiem: Loop
    txtDirDat_P0.Move 7800, 2010
    If icolor1 = 1 Then
      Label5.Caption = "Traida de instruccion"
      Label5.Visible = True
      Label5.ForeColor = QBColor(4)
      txtDirDat_P0.Text = sInstruccionesOperandos(ejec1, 16)
```

```
    txtDirDat_P0.Visible = True
    imgArrow1.Picture = frmImagenes.imgLeftarrow1.Picture
    imgArrow1.Move 8040, 2040
    imgArrow1.Visible = True
  Else
    Label5.Caption = "Lectura no válida"
    Label5.Visible = True
  End If
  P0_Eprom_QBcolor1
  frmDiagrama2.Refresh
  Start = Timer: Do While Timer < Start + 0.7 * tiem: Loop  'Es 0.5 seg en realidad
  txtDirDat_P0.Move 5640, 2660
  imgArrow1.Move 5880, 2690
  frmDiagrama2.Refresh
  Start = Timer: Do While Timer < Start + 0.6 * tiem: Loop  'Es 0.5 seg en realidad
  txtDirDat_P0.Move 2760, 2180
  imgArrow1.Move 2520, 2220
  frmDiagrama2.Refresh
  Start = Timer: Do While Timer < Start + 0.7 * tiem: Loop  'Es 0.5 seg en realidad
  txtDirDat_P0.Visible = False
  imgArrow1.Visible = False
  imgArrow1.Picture = frmImagenes.imgRightarrow1.Picture
  Label4.Visible = False
  Label5.Visible = False
  frmDiagrama2.Refresh
  txtDir_P2.Move 2760, 1190
  txtDirDat_P0.Move 2760, 2180
  imgArrow1.Move 2520, 2220
  If icolor1 = 2 Or icolor1 = 4 Then
    'Solo en la finalización de cada ciclo de maquina
    Form2.Temp_Cont_0
    Form2.Temp_Cont_1
  End If
  Next icolor1
  label1.Visible = False
  Label2.Visible = False
  Label3.Visible = False
  frmDiagrama2.Refresh
  Form2.Atencion_Interrupciones
  End Sub

  Sub Byte1_2Ciclomovc_1()
  EscrituraP1
  'Cada fase tendrá una duración de 2tiem segundo
  tiem = Frecuencia_Oscilador
  icolor = 16777215  'Line1.BorderColor
  icolor2 = 16777215  'Shape1.BackColor
  icolor3 = 8421504  'Shape4.BackColor
  label1.Caption = saMatrizInstrucciones(ejec1)
  Label2.Caption = "#Bytes = " & sInstruccionesOperandos(ejec1, 8)
  Label3.Caption = "#C.Maq = " & sInstruccionesOperandos(ejec1, 9)
  label1.Visible = True
  Label2.Visible = True
  Label3.Visible = True
  For icolor1 = 1 To 4
  If icolor1 = 1 Or icolor1 = 2 Or icolor1 = 4 Then '3  en 2
    txtDir_P2.Text = sInstruccionesOperandos(ejec1, 19)
    txtDirDat_P0.Text = sInstruccionesOperandos(ejec1, 20)
  ElseIf icolor1 = 3 Then '2 en 3
    txtDir_P2.Text = sAddP2
    txtDirDat_P0.Text = sAddP0
  End If
  'Segunda fase primer estado, primera fase segundo estado,
  ALE_Latch_QBcolor1
  PSEN_Eprom_QBcolor1
  P2_Eprom_QBcolor1
  P0_Eprom_color1
  'Para Puerto P1 en WR
    If icolor1 = 1 And ejec1 > 0 And iEscrituraP1 Then
      If sInstruccionesOperandos(ejec1 - 1, 2) = "144" Or Mid(sInstruccionesOperandos(ejec1 - 1, 2), 1,
                                                    2) = "P1" Then
```

```
        Puerto_P1_QBcolor1
      End If
    End If
  frmDiagrama2.Refresh
  Start = Timer: Do While Timer < Start + tiem: Loop
  P2_Eprom_color1
  'Para Puerto P1 en WR
    If icolor1 = 1 And ejec1 > 0 And iEscrituraP1 Then
      If sInstruccionesOperandos(ejec1 - 1, 2) = "144" Then
        txtPuerto_P1.Text = sMemoriaRAMintH(144)
        txtPuerto_P1.Visible = True
        imgArrow_P1.Visible = True
      ElseIf Mid(sInstruccionesOperandos(ejec1 - 1, 2), 1, 2) = "P1" Then
        txtPuerto_P1.ForeColor = QBColor(12)
        txtPuerto_P1.Text =   Mid(sInstruccionesOperandos(ejec1 - 1, 2), 3, 4)
        txtPuerto_P1.Visible = True
        imgArrow_P1.Visible = True
      End If
    End If
  frmDiagrama2.Refresh
  Start = Timer: Do While Timer < Start + tiem: Loop
  'Para Puerto P1 en WR
    If icolor1 = 1 And ejec1 > 0 And iEscrituraP1 Then
      If sInstruccionesOperandos(ejec1 - 1, 2) = "144" Or Mid(sInstruccionesOperandos(ejec1 - 1, 2), 1,
                                                  2) = "P1" Then
        txtPuerto_P1 = ""
        txtPuerto_P1.Visible = False
        imgArrow_P1.Visible = False
        txtPuerto_P1.ForeColor = QBColor(0)
        Form2.Puerto_P1WR
        Puerto_P1_color1
      End If
    End If
  P2_Eprom_QBcolor1
  P0_Latch_QBcolor1
  If icolor1 = 1 Or icolor1 = 2 Or icolor1 = 4 Then '3 en 2
    Label4.Caption = "Direccionamiento para instrucción"
    Label4.ForeColor = QBColor(4)
    Label4.Visible = True
  ElseIf icolor1 = 3 Then '2 en 3
    Label4.Caption = "Direccionamiento para dato"
    Label4.Visible = True
    Label4.ForeColor = QBColor(4)
  End If
  txtDir_P2.Move 2760, 1190
  txtDir_P2.Visible = True
  txtDirDat_P0.Visible = True
  imgArrow1.Visible = True
  imgArrow2.Visible = True
  frmDiagrama2.Refresh
  Start = Timer: Do While Timer < Start + tiem: Loop
  'Envio del P2 y P0 como dirrecciones
  txtDir_P2.Move 6120, 150
  txtDirDat_P0.Move 5520, 1640
  imgArrow1.Move 5760, 1680
  imgArrow2.Move 5880, 160
  frmDiagrama2.Refresh
  Start = Timer: Do While Timer < Start + tiem: Loop
  'Segunda fase segundo estado
  ALE_Latch_color1
  Latch_Eprom_QBcolor1
  txtDir_P2.Move 9480, 150
  txtDirDat_P0.Move 7320, 1640
  imgArrow1.Move 7080, 1680
  imgArrow2.Move 9240, 160
  frmDiagrama2.Refresh
  Start = Timer: Do While Timer < Start + tiem: Loop
  txtDir_P2.Move 9600, 630
  txtDirDat_P0.Move 7800, 920
  imgArrow1.Move 8040, 960
  imgArrow2.Picture = frmImagenes.imgLeftarrow1.Picture
```

```
imgArrow2.Move 9360, 660
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
txtDir_P2.Visible = False
txtDirDat_P0.Visible = False
imgArrow1.Visible = False
imgArrow2.Visible = False
imgArrow2.Picture = frmImagenes.imgRightarrow1.Picture
frmDiagrama2.Refresh
'Primera fase tercer estado, segunda fase tercer estado,
'primera fase cuarto estado
PSEN_Eprom_color1
P0_Latch_color1
Latch_Eprom_color1
Label4.ForeColor = QBColor(0)
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 4 * tiem: Loop
txtDirDat_P0.Move 7800, 2010
imgArrow1.Picture = frmImagenes.imgLeftarrow1.Picture
imgArrow1.Move 8040, 2040
If icolor1 = 1 Then
   txtDirDat_P0.Text = sInstruccionesOperandos(ejec1, 16)
   txtDirDat_P0.Visible = True
   imgArrow1.Visible = True
   Label5.Caption = "Traida de instrucción"
ElseIf icolor1 = 3 Then     '2 en 3
   txtDirDat_P0.Text = sDatP0
   txtDirDat_P0.Visible = True
   imgArrow1.Visible = True
   Label5.Caption = "Traida de dato"
Else
   Label5.Caption = "Lectura no válida"
End If
   Label5.ForeColor = QBColor(4)
   Label5.Visible = True
P0_Eprom_QBcolor1
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 0.7 * tiem: Loop 'Es 0.5 seg en realidad
txtDirDat_P0.Move 5640, 2660
imgArrow1.Move 5880, 2690
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 0.6 * tiem: Loop  'Es 0.5 seg en realidad
txtDirDat_P0.Move 2760, 2180
imgArrow1.Move 2520, 2220
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 0.7 * tiem: Loop  'Es 0.5 seg en realidad
Label5.Visible = False
Label5.ForeColor = QBColor(0)
Label4.Visible = False
txtDirDat_P0.Visible = False
imgArrow1.Visible = False
txtDir_P2.Move 2760, 1190
txtDirDat_P0.Move 2760, 2180
imgArrow1.Picture = frmImagenes.imgRightarrow1.Picture
imgArrow2.Move 2520, 1200
frmDiagrama2.Refresh
If icolor1 = 2 Or icolor1 = 4 Then
   'Solo en la finalización de los ciclos de maquina
   Form2.Temp_Cont_0
   Form2.Temp_Cont_1
End If
Next icolor1
label1.Visible = False
Label2.Visible = False
Label3.Visible = False
frmDiagrama2.Refresh
Form2.Atencion_Interrupciones
End Sub

Sub Byte1_2Ciclomovc_NoMOVC()
EscrituraP1
```

```
'Cada fase tendrá una duración de 2tiem segundo
tiem = Frecuencia_Oscilador
icolor = 16777215  'Line1.BorderColor
icolor2 = 16777215  'Shape1.BackColor
icolor3 = 8421504  'Shape4.BackColor
label1.Caption = saMatrizInstrucciones(ejec1)
Label2.Caption = "#Bytes = " & sInstruccionesOperandos(ejec1, 8)
Label3.Caption = "#C.Maq = " & sInstruccionesOperandos(ejec1, 9)
label1.Visible = True
Label2.Visible = True
Label3.Visible = True
For icolor1 = 1 To 4
If icolor1 = 1 Or icolor1 = 2 Or icolor1 = 4 Then '3  en 2
   txtDir_P2.Text = sInstruccionesOperandos(ejec1, 19)
   txtDirDat_P0.Text = sInstruccionesOperandos(ejec1, 20)
ElseIf icolor1 = 3 Then '2 en 3
   txtDir_P2.Text = sAddP2
   txtDirDat_P0.Text = sAddP0
End If
'Segunda fase primer estado, primera fase segundo estado,
ALE_Latch_QBcolor1
PSEN_Eprom_QBcolor1
P2_Eprom_QBcolor1
P0_Eprom_color1
'Para Puerto P1 en WR
   If icolor1 = 1 And ejec1 > 0 And iEscrituraP1 Then
      If sInstruccionesOperandos(ejec1 - 1, 2) = "144" Or Mid(sInstruccionesOperandos(ejec1 - 1, 2), 1,
                                                 2) = "P1" Then
         Puerto_P1_QBcolor1
      End If
   End If
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
P2_Eprom_color1
'Para Puerto P1 en WR
   If icolor1 = 1 And ejec1 > 0 And iEscrituraP1 Then
      If sInstruccionesOperandos(ejec1 - 1, 2) = "144" Then
         txtPuerto_P1.Text = sMemoriaRAMintH(144)
         txtPuerto_P1.Visible = True
         imgArrow_P1.Visible = True
      ElseIf Mid(sInstruccionesOperandos(ejec1 - 1, 2), 1, 2) = "P1" Then
         txtPuerto_P1.ForeColor = QBColor(12)
         txtPuerto_P1.Text = Mid(sInstruccionesOperandos(ejec1 - 1, 2), 3, 4)
         txtPuerto_P1.Visible = True
         imgArrow_P1.Visible = True
      End If
   End If
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
'Para Puerto P1 en WR
   If icolor1 = 1 And ejec1 > 0 And iEscrituraP1 Then
      If sInstruccionesOperandos(ejec1 - 1, 2) = "144" Or Mid(sInstruccionesOperandos(ejec1 - 1, 2), 1,
                                                 2) = "P1" Then
         txtPuerto_P1 = ""
         txtPuerto_P1.Visible = False
         imgArrow_P1.Visible = False
         txtPuerto_P1.ForeColor = QBColor(0)
         Form2.Puerto_P1WR
         Puerto_P1_color1
      End If
   End If
P2_Eprom_QBcolor1
P0_Latch_QBcolor1
If icolor1 = 1 Or icolor1 = 2 Or icolor1 = 4 Then
   Label4.Caption = "Direccionamiento para instrucción"
   Label4.ForeColor = QBColor(4)
   Label4.Visible = True
ElseIf icolor1 = 3 Then '2 en 3
   Label4.Caption = "Direccionamiento para dato"
   Label4.Visible = True
   Label4.ForeColor = QBColor(4)
```

```
    lblDecoder.Caption = iRomc1
    lblDec_Salida.Caption = "Ninguna"
    Label14.Visible = False
    Line60.Visible = False
    Select Case iRomc1
      Case 1
        Line53.Y1 = 1515
        Line53.Y2 = 1515
        lblDec_Salida.Top = 1425
      Case 2
        Line53.Y1 = 1605
        Line53.Y2 = 1605
        lblDec_Salida.Top = 1515
      Case 3
        Line53.Y1 = 1710
        Line53.Y2 = 1710
        lblDec_Salida.Top = 1620
      Case 4
        Line53.Y1 = 1800
        Line53.Y2 = 1800
        lblDec_Salida.Top = 1710
      Case 5
        Line53.Y1 = 1905
        Line53.Y2 = 1905
        lblDec_Salida.Top = 1815
      Case 6
        Line53.Y1 = 1995
        Line53.Y2 = 1995
        lblDec_Salida.Top = 1905
      Case 7
        Line53.Y1 = 2085
        Line53.Y2 = 2085
        lblDec_Salida.Top = 1995
    End Select
  End If
  txtDir_P2.Move 2760, 1190
  txtDir_P2.Visible = True
  txtDirDat_P0.Visible = True
  imgArrow1.Visible = True
  imgArrow2.Visible = True
  frmDiagrama2.Refresh
  Start = Timer: Do While Timer < Start + tiem: Loop
  'Envio del P2 y P0 como dirrecciones
  txtDir_P2.Move 6120, 150
  txtDirDat_P0.Move 5520, 1640
  imgArrow1.Move 5760, 1680
  imgArrow2.Move 5880, 160
  frmDiagrama2.Refresh
  Start = Timer: Do While Timer < Start + tiem: Loop
  'Segunda fase segundo estado
  ALE_Latch_color1
  Latch_Eprom_QBcolor1
  txtDir_P2.Move 9480, 150
  txtDirDat_P0.Move 7320, 1640
  imgArrow1.Move 7080, 1680
  imgArrow2.Move 9240, 160
  frmDiagrama2.Refresh
  Start = Timer: Do While Timer < Start + tiem: Loop
  txtDir_P2.Move 9600, 630
  txtDirDat_P0.Move 7800, 920
  imgArrow1.Move 8040, 960
  imgArrow2.Picture = frmImagenes.imgLeftarrow1.Picture
  imgArrow2.Move 9360, 660
  frmDiagrama2.Refresh
  Start = Timer: Do While Timer < Start + tiem: Loop
  txtDir_P2.Visible = False
  txtDirDat_P0.Visible = False
  imgArrow1.Visible = False
  imgArrow2.Visible = False
  imgArrow2.Picture = frmImagenes.imgRightarrow1.Picture
  frmDiagrama2.Refresh
```

```
'Primera fase tercer estado, segunda fase tercer estado,
'primera fase cuarto estado
PSEN_Eprom_color1
P0_Latch_color1
Latch_Eprom_color1
Label4.ForeColor = QBColor(0)
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 4 * tiem: Loop
txtDirDat_P0.Move 7800, 2010
imgArrow1.Picture = frmImagenes.imgLeftarrow1.Picture
imgArrow1.Move 8040, 2040
If icolor1 = 1 Then
   txtDirDat_P0.Text = sInstruccionesOperandos(ejec1, 16)
   txtDirDat_P0.Visible = True
   imgArrow1.Visible = True
   Label5.Caption = "Traida de instrucción"
ElseIf icolor1 = 3 Then
   txtDirDat_P0.Move 7200, 2660
   imgArrow1.Move 7485, 2690
   txtDirDat_P0.Text = "FF"
   txtDirDat_P0.Visible = True
   imgArrow1.Visible = True
   Label5.Caption = "Traida de dato"
Else
   Label5.Caption = "Lectura no válida"
End If
   Label5.ForeColor = QBColor(4)
   Label5.Visible = True
If icolor1 = 3 Then
   P0_NoMOVC_QBcolor1
   Label10.Caption = "de dónde?"
   Label10.Top = 7920
   Label10.Left = 2700
   Label10.Visible = True
Else
   P0_Eprom_QBcolor1
End If
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 0.7 * tiem: Loop 'Es 0.5 seg en realidad
txtDirDat_P0.Move 5640, 2660
imgArrow1.Move 5880, 2690
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 0.6 * tiem: Loop  'Es 0.5 seg en realidad
txtDirDat_P0.Move 2760, 2180
imgArrow1.Move 2520, 2220
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 0.7 * tiem: Loop  'Es 0.5 seg en realidad
Label5.Visible = False
Label5.ForeColor = QBColor(0)
Label4.Visible = False
txtDirDat_P0.Visible = False
imgArrow1.Visible = False
txtDir_P2.Move 2760, 1190
txtDirDat_P0.Move 2760, 2180
imgArrow1.Picture = frmImagenes.imgRightarrow1.Picture
imgArrow2.Move 2520, 1200
frmDiagrama2.Refresh
If icolor1 = 2 Or icolor1 = 4 Then
   'Solo en la finalización de los ciclos de maquina
   Form2.Temp_Cont_0
   Form2.Temp_Cont_1
End If
If icolor1 = 3 Then
   Label10.Visible = False
   Label10.Top = 6240
   Label10.Left = 3840
End If
Next icolor1
label1.Visible = False
Label2.Visible = False
Label3.Visible = False
```

```
'P2_Eprom_QBcolor1
frmDiagrama2.Refresh
Form2.Atencion_Interrupciones
End Sub

Sub Byte1_2CiclomovxR_1()
EscrituraP1
'Cada fase tendrá una duración de 2tiem segundo
tiem = Frecuencia_Oscilador
icolor = 16777215   'Line1.BorderColor
icolor2 = 16777215  'Shape1.BackColor
icolor3 = 8421504   'Shape4.BackColor
label1.Caption = saMatrizInstrucciones(ejec1)
Label2.Caption = '#Bytes = " & sInstruccionesOperandos(ejec1, 8)
Label3.Caption = '#C.Maq = " & sInstruccionesOperandos(ejec1, 9)
label1.Visible = True
Label2.Visible = True
Label3.Visible = True
txtDir_P2.Text = sInstruccionesOperandos(ejec1, 19)
txtDirDat_P0.Text = sInstruccionesOperandos(ejec1, 20)
'Segunda fase primer estado, primera fase segundo estado,
ALE_Latch_QBcolor1
PSEN_Eprom_QBcolor1
P2_Eprom_QBcolor1
P0_Eprom_color1
'Para Puerto P1 en WR o RD
   If ejec1 > 0 And iEscrituraP1 Then
      If sInstruccionesOperandos(ejec1 - 1, 2) = "144" Or Mid(sInstruccionesOperandos(ejec1 - 1, 2), 1,
                                                 2) = "P1" Then
         Puerto_P1_QBcolor1
      End If
   End If
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
P2_Eprom_color1
'Para Puerto P1 en WR o RD
   If ejec1 > 0 And iEscrituraP1 Then      If sInstruccionesOperandos(ejec1 - 1, 2) = "144" Then
         txtPuerto_P1.Text = sMemoriaRAMintH(144)
         imgArrow_P1.Picture = frmImagenes.imgRightarrow1.Picture
         txtPuerto_P1.Visible = True
         imgArrow_P1.Visible = True
      ElseIf Mid(sInstruccionesOperandos(ejec1 - 1, 2), 1, 2) = "P1" Then
         txtPuerto_P1.ForeColor = QBColor(12)
         txtPuerto_P1.Text = Mid(sInstruccionesOperandos(ejec1 - 1, 2), 3, 4)
         txtPuerto_P1.Visible = True
         imgArrow_P1.Visible = True
      End If
   End If
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
'Para Puerto P1 en WR
   If ejec1 > 0 And iEscrituraP1 Then
      If sInstruccionesOperandos(ejec1 - 1, 2) = "144" Or Mid(sInstruccionesOperandos(ejec1 - 1, 2), 1,
                                                 2) = "P1" Then
         txtPuerto_P1 = ""
         txtPuerto_P1.Visible = False
         imgArrow_P1.Visible = False
         txtPuerto_P1.ForeColor = QBColor(0)
         Form2.Puerto_P1WR
         Puerto_P1_color1
      End If
   End If
P2_Eprom_QBcolor1
P0_Latch_QBcolor1
Label4.ForeColor = QBColor(4)
Label4.Caption = "Direccionamiento a ROM"
Label4.Visible = True
txtDir_P2.Visible = True
txtDirDat_P0.Visible = True
imgArrow1.Visible = True
imgArrow2.Visible = True
```

```
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
'Envio del P2 y P0 como direcciones
txtDir_P2.Move 6120, 150
txtDirDat_P0.Move 5520, 1640
imgArrow1.Move 5760, 1680
imgArrow2.Move 5880, 160
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
'Segunda fase segundo estado
ALE_Latch_color1
PSEN_Eprom_QBcolor1
Latch_Eprom_QBcolor1
txtDir_P2.Move 9480, 150
txtDirDat_P0.Move 7320, 1640
imgArrow1.Move 7080, 1680
imgArrow2.Move 9240, 160
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
txtDir_P2.Move 9600, 630
txtDirDat_P0.Move 7800, 920
imgArrow1.Move 8040, 960
imgArrow2.Picture = frmImagenes.imgLeftarrow1.Picture
imgArrow2.Move 9360, 660
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
txtDir_P2.Visible = False
txtDirDat_P0.Visible = False
imgArrow1.Visible = False
imgArrow2.Visible = False
imgArrow2.Picture = frmImagenes.imgRightarrow1.Picture
frmDiagrama2.Refresh
'Primera fase tercer estado, segunda fase tercer estado,
'primera fase cuarto estado
ALE_Latch_color1
PSEN_Eprom_color1
P0_Latch_color1
Latch_Eprom_color1
Label4.ForeColor = QBColor(0)
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 4 * tiem: Loop
txtDirDat_P0.Move 7800, 2010
imgArrow1.Picture = frmImagenes.imgLeftarrow1.Picture
imgArrow1.Move 8040, 2040
Label5.ForeColor = QBColor(4)
Label5.Caption = "Lectura de instrucción"
Label5.Visible = True
txtDirDat_P0.Text = sInstruccionesOperandos(ejec1, 16)
txtDirDat_P0.Visible = True
imgArrow1.Visible = True
P0_Eprom_QBcolor1
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 0.7 * tiem: Loop  'Es 0.5 seg en realidad
txtDirDat_P0.Move 5640, 2660
imgArrow1.Move 5880, 2690
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 0.6 * tiem: Loop  'Es 0.5 seg en realidad
txtDirDat_P0.Move 2760, 2180
imgArrow1.Move 2520, 2220
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 0.7 * tiem: Loop  'Es 0.5 seg en realidad
Label4.Visible = False
Label5.Visible = False
txtDirDat_P0.Visible = False
imgArrow1.Visible = False
P0_Eprom_color1
frmDiagrama2.Refresh
txtDirDat_P0.Move 2760, 2180
txtDir_P2.Move 2760, 1190
imgArrow1.Picture = frmImagenes.imgRightarrow1.Picture
imgArrow2.Move 2520, 1200
```

```
'S4P2
ALE_Latch_QBcolor1
PSEN_Eprom_QBcolor1
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
P2_Eprom_color1
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
'S5P1
Label4.ForeColor = QBColor(4)
Label4.Caption = "Direccionamiento a RAM"
Label4.Visible = True
P2_Ram_QBcolor1
P0_Latch_QBcolor1
txtDir_P2.Text = sAddP2
txtDirDat_P0.Text = sAddP0
txtDir_P2.Visible = True
txtDirDat_P0.Visible = True
imgArrow1.Visible = True
imgArrow2.Visible = True
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
txtDir_P2.Move 7440, 150
txtDirDat_P0.Move 5520, 1640
imgArrow1.Move 5760, 1680
imgArrow2.Move 7200, 160
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
'S5P2
ALE_Latch_color1
Latch_Ram_QBcolor1
txtDir_P2.Move 9890, 1440
txtDirDat_P0.Move 7320, 1640
imgArrow1.Move 7080, 1680
imgArrow2.Move 9900, 1200
imgArrow2.Picture = frmImagenes.imgDownarrow1.Picture
imgArrow2.Height = 255
imgArrow2.Width = 150
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 3 * tiem: Loop
txtDir_P2.Move 9360, 3360
txtDirDat_P0.Move 7560, 3690
imgArrow1.Move 7800, 3720
imgArrow2.Move 9120, 3390
imgArrow2.Picture = frmImagenes.imgLeftarrow1.Picture
imgArrow2.Height = 150
imgArrow2.Width = 255
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 3 * tiem: Loop
'Ya estan incluidos arriba S6P1,S6P2,
txtDir_P2.Visible = False
txtDirDat_P0.Visible = False
imgArrow1.Visible = False
imgArrow2.Visible = False
txtDir_P2.Move 2760, 1190
txtDirDat_P0.Move 2760, 2180
imgArrow1.Move 2520, 2220
imgArrow2.Picture = frmImagenes.imgRightarrow1.Picture
imgArrow2.Move 2520, 1200
'CICLO 2:
'S1P1, S1P2
Label4.ForeColor = QBColor(0)
RD_Ram_QBcolor1
P0_Latch_color1
Latch_Ram_color1
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 2 * tiem: Loop
'Finaliza primer ciclo de maquina
Form2.Temp_Cont_0
Form2.Temp_Cont_1
Start = Timer: Do While Timer < Start + 4 * tiem: Loop
```

```
'CICLO 2 = S1P1 hasta S2P1
'CICLO 2
'S2P2, S3P1, S3P2
Label5.Caption = "Lectura de RAM"
Label5.Visible = True
P0_Ram_QBcolor1
txtDirDat_P0.Text = sDatP0
txtDirDat_P0.Move 7560, 4770
imgArrow1.Move 7800, 4800
imgArrow1.Picture = frmImagenes.imgLeftarrow1.Picture
txtDirDat_P0.Visible = True
imgArrow1.Visible = True
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 1.5 * tiem: Loop
txtDirDat_P0.Move 7060, 3240
imgArrow1.Move 7140, 3480
imgArrow1.Picture = frmImagenes.imgUparrow1.Picture
imgArrow1.Height = 255
imgArrow1.Width = 150
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 1.5 * tiem: Loop
txtDirDat_P0.Move 5640, 2660
imgArrow1.Move 5880, 2690
imgArrow1.Picture = frmImagenes.imgLeftarrow1.Picture
imgArrow1.Height = 150
imgArrow1.Width = 255
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 1.5 * tiem: Loop
txtDirDat_P0.Move 2760, 2180
imgArrow1.Move 2520, 2220
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 1.5 * tiem: Loop
Label4.Visible = False
Label5.Visible = False
'S4P1
txtDirDat_P0.Visible = False
imgArrow1.Visible = False
imgArrow1.Picture = frmImagenes.imgRightarrow1.Picture
RD_Ram_color1
P0_Ram_color1
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 2 * tiem: Loop
'S4P2
ALE_Latch_QBcolor1
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
P2_Ram_color1
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
'S5P1
txtDir_P2.Text = sInstruccionesOperandos(ejec1, 19)
txtDirDat_P0.Text = sInstruccionesOperandos(ejec1, 20)
txtDir_P2.Visible = True
txtDirDat_P0.Visible = True
imgArrow1.Visible = True
imgArrow2.Visible = True
Label4.ForeColor = QBColor(4)
Label4.Caption = "Direccionamiento a ROM"
Label4.Visible = True
P2_Eprom_QBcolor1
P0_Latch_QBcolor1
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
'Envio del P2 y P0 como direcciones
txtDir_P2.Move 6120, 150
txtDirDat_P0.Move 5520, 1640
imgArrow1.Move 5760, 1680
imgArrow2.Move 5880, 160
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
'Segunda fase segundo estado
```

```
ALE_Latch_color1
PSEN_Eprom_QBcolor1
Latch_Eprom_QBcolor1
txtDir_P2.Move 9480, 150
txtDirDat_P0.Move 7320, 1640
imgArrow1.Move 7080, 1680
imgArrow2.Move 9240, 160
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
txtDir_P2.Move 9600, 630
txtDirDat_P0.Move 7800, 920
imgArrow1.Move 8040, 960
imgArrow2.Picture = frmImagenes.imgLeftarrow1.Picture
imgArrow2.Move 9360, 660
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
txtDir_P2.Visible = False
txtDirDat_P0.Visible = False
imgArrow1.Visible = False
imgArrow2.Visible = False
imgArrow2.Picture = frmImagenes.imgRightarrow1.Picture
frmDiagrama2.Refresh
'Primera fase tercer estado, segunda fase tercer estado,
'primera fase cuarto estado
ALE_Latch_color1
PSEN_Eprom_color1
P0_Latch_color1
Latch_Eprom_color1
Label4.ForeColor = QBColor(0)
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 4 * tiem: Loop
Label5.ForeColor = QBColor(4)
Label5.Caption = "Lectura no válida"
Label5.Visible = True
P0_Eprom_QBcolor1
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 0.7 * tiem: Loop  'Es 0.5 seg en realidad
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 0.6 * tiem: Loop  'Es 0.5 seg en realidad
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 0.7 * tiem: Loop  'Es 0.5 seg en realidad
label1.Visible = False
Label2.Visible = False
Label3.Visible = False
Label4.Visible = False
Label5.Visible = False
'Finaliza segundo ciclo de maquina
Form2.Temp_Cont_0
Form2.Temp_Cont_1
'S1P2
txtDir_P2.Move 2760, 1190
txtDirDat_P0.Move 2760, 2180
imgArrow1.Move 2520, 2220
imgArrow2.Move 2520, 1200
frmDiagrama2.Refresh
Form2.Atencion_Interrupciones
End Sub

Sub Byte1_2CiclomovxR_Buffer()
EscrituraP1
'Cada fase tendrá una duración de 2tiem segundo
tiem = Frecuencia_Oscilador
icolor = 16777215   'Line1.BorderColor
icolor2 = 16777215  'Shape1.BackColor
icolor3 = 8421504   'Shape4.BackColor
label1.Caption = saMatrizInstrucciones(ejec1)
Label2.Caption = "#Bytes = " & sInstruccionesOperandos(ejec1, 8)
Label3.Caption = "#C.Maq = " & sInstruccionesOperandos(ejec1, 9)
label1.Visible = True
Label2.Visible = True
Label3.Visible = True
```

```
txtDir_P2.Text = sInstruccionesOperandos(ejec1, 19)
txtDirDat_P0.Text = sInstruccionesOperandos(ejec1, 20)
'Segunda fase primer estado, primera fase segundo estado,
ALE_Latch_QBcolor1
PSEN_Eprom_QBcolor1
P2_Eprom_QBcolor1
P0_Eprom_color1
'Para Puerto P1 en WR
  If ejec1 > 0 And iEscrituraP1 Then
    If sInstruccionesOperandos(ejec1 - 1, 2) = "144" Or Mid(sInstruccionesOperandos(ejec1 - 1, 2), 1,
                                                2) = "P1" Then
      Puerto_P1_QBcolor1
    End If
  End If
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
P2_Eprom_color1
'Para Puerto P1 en WR
  If ejec1 > 0 And iEscrituraP1 Then
    If sInstruccionesOperandos(ejec1 - 1, 2) = "144" Then
      txtPuerto_P1.Text = sMemoriaRAMintH(144)
      txtPuerto_P1.Visible = True
      imgArrow_P1.Visible = True
    ElseIf Mid(sInstruccionesOperandos(ejec1 - 1, 2), 1, 2) = "P1" Then
      txtPuerto_P1.ForeColor = QBColor(12)
      txtPuerto_P1.Text = Mid(sInstruccionesOperandos(ejec1 - 1, 2), 3, 4)
      txtPuerto_P1.Visible = True
      imgArrow_P1.Visible = True
    End If
  End If
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
'Para Puerto P1 en WR
  If ejec1 > 0 And iEscrituraP1 Then
    If sInstruccionesOperandos(ejec1 - 1, 2) = "144" Or Mid(sInstruccionesOperandos(ejec1 - 1, 2), 1,
                                                2) = "P1" Then
      txtPuerto_P1 = ""
      txtPuerto_P1.Visible = False
      txtPuerto_P1.ForeColor = QBColor(0)
      imgArrow_P1.Visible = False
      Form2.Puerto_P1WR
      Puerto_P1_color1
    End If
  End If
P2_Eprom_QBcolor1
P0_Latch_QBcolor1
Label4.ForeColor = QBColor(4)
Label4.Caption = "Direccionamiento a ROM"
Label4.Visible = True
txtDir_P2.Visible = True
txtDirDat_P0.Visible = True
imgArrow1.Visible = True
imgArrow2.Visible = True
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
'Envio del P2 y P0 como direcciones
txtDir_P2.Move 6120, 150
txtDirDat_P0.Move 5520, 1640
imgArrow1.Move 5760, 1680
imgArrow2.Move 5880, 160
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
'Segunda fase segundo estado
ALE_Latch_color1
PSEN_Eprom_QBcolor1
Latch_Eprom_QBcolor1
txtDir_P2.Move 9480, 150
txtDirDat_P0.Move 7320, 1640
imgArrow1.Move 7080, 1680
imgArrow2.Move 9240, 160
frmDiagrama2.Refresh
```

```
Start = Timer: Do While Timer < Start + tiem: Loop
txtDir_P2.Move 9600, 630
txtDirDat_P0.Move 7800, 920
imgArrow1.Move 8040, 960
imgArrow2.Picture = frmImagenes.imgLeftarrow1.Picture
imgArrow2.Move 9360, 660
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
txtDir_P2.Visible = False
txtDirDat_P0.Visible = False
imgArrow1.Visible = False
imgArrow2.Visible = False
imgArrow2.Picture = frmImagenes.imgRightarrow1.Picture
frmDiagrama2.Refresh
'Primera fase tercer estado, segunda fase tercer estado,
'primera fase cuarto estado
ALE_Latch_color1
PSEN_Eprom_color1
P0_Latch_color1
Latch_Eprom_color1
Label4.ForeColor = QBColor(0)
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 4 * tiem: Loop
txtDirDat_P0.Move 7800, 2010
imgArrow1.Picture = frmImagenes.imgLeftarrow1.Picture
imgArrow1.Move 8040, 2040
Label5.ForeColor = QBColor(4)
Label5.Caption = "Lectura de instrucción"
Label5.Visible = True
txtDirDat_P0.Text = sInstruccionesOperandos(ejec1, 16)
txtDirDat_P0.Visible = True
imgArrow1.Visible = True
P0_Eprom_QBcolor1
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 0.7 * tiem: Loop  'Es 0.5 seg en realidad
txtDirDat_P0.Move 5640, 2660
imgArrow1.Move 5880, 2690
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 0.6 * tiem: Loop  'Es 0.5 seg en realidad
txtDirDat_P0.Move 2760, 2180
imgArrow1.Move 2520, 2220
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 0.7 * tiem: Loop  'Es 0.5 seg en realidad
Label4.Visible = False
Label5.Visible = False
txtDirDat_P0.Visible = False
imgArrow1.Visible = False
P0_Eprom_color1
frmDiagrama2.Refresh
txtDirDat_P0.Move 2760, 2180
txtDir_P2.Move 2760, 1190
imgArrow1.Picture = frmImagenes.imgRightarrow1.Picture
imgArrow2.Move 2520, 1200
'S4P2
ALE_Latch_QBcolor1
PSEN_Eprom_QBcolor1
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
P2_Eprom_color1
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
'S5P1
Label4.ForeColor = QBColor(4)
Label4.Caption = "Direccionamiento a RAM"
Label4.Visible = True
P2_Ram_QBcolor1
P0_Latch_QBcolor1
txtDir_P2.Text = sAddP2
txtDirDat_P0.Text = sAddP0
txtDir_P2.Visible = True
txtDirDat_P0.Visible = True
```

```
imgArrow1.Visible = True
imgArrow2.Visible = True
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
txtDir_P2.Move 7440, 150
txtDirDat_P0.Move 5520, 1640
imgArrow1.Move 5760, 1680
imgArrow2.Move 7200, 160
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
'S5P2
ALE_Latch_color1
Latch_Ram_QBcolor1
txtDir_P2.Move 9890, 1440
txtDirDat_P0.Move 7320, 1640
imgArrow1.Move 7080, 1680
imgArrow2.Move 9900, 1200
imgArrow2.Picture = frmImagenes.imgDownarrow1.Picture
imgArrow2.Height = 255
imgArrow2.Width = 150
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 3 * tiem: Loop
txtDir_P2.Move 9360, 3360
txtDirDat_P0.Move 7560, 3690
imgArrow1.Move 7800, 3720
imgArrow2.Move 9120, 3390
imgArrow2.Picture = frmImagenes.imgLeftarrow1.Picture
imgArrow2.Height = 150
imgArrow2.Width = 255
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 3 * tiem: Loop
'Ya estan incluidos arriba S6P1,S6P2,
txtDir_P2.Visible = False
txtDirDat_P0.Visible = False
imgArrow1.Visible = False
imgArrow2.Visible = False
txtDir_P2.Move 2760, 1190
txtDirDat_P0.Move 2760, 2180
imgArrow1.Move 2520, 2220
imgArrow2.Picture = frmImagenes.imgRightarrow1.Picture
imgArrow2.Move 2520, 1200
'CICLO 2:
'S1P1, S1P2
Label4.ForeColor = QBColor(0)
RD_Buffer_QBcolor1
P0_Latch_color1
Latch_Ram_color1
txtDirDat_P0.Text = sValor_InH_Buffer
txtDirDat_P0.Move 3600, 4095  '3390, 4095
imgArrow1.Move 3870, 4125 '3660, 4125
imgArrow1.Picture = frmImagenes.imgRightarrow1.Picture
txtDirDat_P0.Visible = True
imgArrow1.Visible = True
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 2 * tiem: Loop
'Finaliza primer ciclo de maquina
Form2.Temp_Cont_0
Form2.Temp_Cont_1
Start = Timer: Do While Timer < Start + 4 * tiem: Loop
'CICLO 2 = S1P1 hasta S2P1
'CICLO 2
'S2P2, S3P1, S3P2
Label5.Caption = "Lectura de RAM"
Label5.Visible = True
P0_Buffer_QBcolor1
txtDirDat_P0.Move 5445, 4080
imgArrow1.Move 5175, 4110
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 1.5 * tiem: Loop
txtDirDat_P0.Move 7060, 3240
imgArrow1.Move 7140, 3480
```

```
imgArrow1.Picture = frmImagenes.imgUparrow1.Picture
imgArrow1.Height = 255
imgArrow1.Width = 150
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 1.5 * tiem: Loop
txtDirDat_P0.Move 5640, 2660
imgArrow1.Move 5880, 2690
imgArrow1.Picture = frmImagenes.imgLeftarrow1.Picture
imgArrow1.Height = 150
imgArrow1.Width = 255
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 1.5 * tiem: Loop
txtDirDat_P0.Move 2760, 2180
imgArrow1.Move 2520, 2220
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 1.5 * tiem: Loop
Label4.Visible = False
Label5.Visible = False
'S4P1
txtDirDat_P0.Visible = False
imgArrow1.Visible = False
imgArrow1.Picture = frmImagenes.imgRightarrow1.Picture
RD_Buffer_color1
P0_Buffer_color1
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 2 * tiem: Loop
'S4P2
ALE_Latch_QBcolor1
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
P2_Ram_color1
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
'S5P1
txtDir_P2.Text = sInstruccionesOperandos(ejec1, 19)
txtDirDat_P0.Text = sInstruccionesOperandos(ejec1, 20)
txtDir_P2.Visible = True
txtDirDat_P0.Visible = True
imgArrow1.Visible = True
imgArrow2.Visible = True
Label4.ForeColor = QBColor(4)
Label4.Caption = "Direccionamiento a ROM"
Label4.Visible = True
P2_Eprom_QBcolor1
P0_Latch_QBcolor1
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
'Envio del P2 y P0 como direcciones
txtDir_P2.Move 6120, 150
txtDirDat_P0.Move 5520, 1640
imgArrow1.Move 5760, 1680
imgArrow2.Move 5880, 160
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
'Segunda fase segundo estado
ALE_Latch_color1
PSEN_Eprom_QBcolor1
Latch_Eprom_QBcolor1
txtDir_P2.Move 9480, 150
txtDirDat_P0.Move 7320, 1640
imgArrow1.Move 7080, 1680
imgArrow2.Move 9240, 160
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
txtDir_P2.Move 9600, 630
txtDirDat_P0.Move 7800, 920
imgArrow1.Move 8040, 960
imgArrow2.Picture = frmImagenes.imgLeftarrow1.Picture
imgArrow2.Move 9360, 660
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
```

```
txtDir_P2.Visible = False
txtDirDat_P0.Visible = False
imgArrow1.Visible = False
imgArrow2.Visible = False
imgArrow2.Picture = frmImagenes.imgRightarrow1.Picture
frmDiagrama2.Refresh
'Primera fase tercer estado, segunda fase tercer estado,
'primera fase cuarto estado
ALE_Latch_color1
PSEN_Eprom_color1
P0_Latch_color1
Latch_Eprom_color1
Label4.ForeColor = QBColor(0)
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 4 * tiem: Loop
Label5.ForeColor = QBColor(4)
Label5.Caption = "Lectura no válida"
Label5.Visible = True
P0_Eprom_QBcolor1
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 0.7 * tiem: Loop  'Es 0.5 seg en realidad
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 0.6 * tiem: Loop  'Es 0.5 seg en realidad
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 0.7 * tiem: Loop  'Es 0.5 seg en realidad
label1.Visible = False
Label2.Visible = False
Label3.Visible = False
Label4.Visible = False
Label5.Visible = False
'Finaliza segundo ciclo de maquina
Form2.Temp_Cont_0
Form2.Temp_Cont_1
'S1P2
txtDir_P2.Move 2760, 1190
txtDirDat_P0.Move 2760, 2180
imgArrow1.Move 2520, 2220
imgArrow2.Move 2520, 1200
frmDiagrama2.Refresh
Form2.Atencion_Interrupciones
End Sub

Sub Byte1_2CiclomovxR_NoAdd()
EscrituraP1
'Cada fase tendrá una duración de 2tiem segundo
tiem = Frecuencia_Oscilador
icolor = 16777215   'Line1.BorderColor
icolor2 = 16777215  'Shape1.BackColor
icolor3 = 8421504   'Shape4.BackColor
label1.Caption = saMatrizInstrucciones(ejec1)
Label2.Caption = "#Bytes = " & sInstruccionesOperandos(ejec1, 8)
Label3.Caption = "#C.Maq = " & sInstruccionesOperandos(ejec1, 9)
label1.Visible = True
Label2.Visible = True
Label3.Visible = True
txtDir_P2.Text = sInstruccionesOperandos(ejec1, 19)
txtDirDat_P0.Text = sInstruccionesOperandos(ejec1, 20)
'Segunda fase primer estado, primera fase segundo estado,
ALE_Latch_QBcolor1
PSEN_Eprom_QBcolor1
P2_Eprom_QBcolor1
P0_Eprom_color1
'Para Puerto P1 en WR
   If ejec1 > 0 And iEscrituraP1 Then
      If sInstruccionesOperandos(ejec1 - 1, 2) = "144" Or Mid(sInstruccionesOperandos(ejec1 - 1, 2), 1,
                                                  2) = "P1" Then
         Puerto_P1_QBcolor1
      End If
   End If
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
```

```
P2_Eprom_color1
'Para Puerto P1 en WR
  If ejec1 > 0 And iEscrituraP1 Then
    If sInstruccionesOperandos(ejec1 - 1, 2) = "144" Then
      txtPuerto_P1.Text = sMemoriaRAMintH(144)
      txtPuerto_P1.Visible = True
      imgArrow_P1.Visible = True
    ElseIf Mid(sInstruccionesOperandos(ejec1 - 1, 2), 1, 2) = "P1" Then
      txtPuerto_P1.ForeColor = QBColor(12)
      txtPuerto_P1.Text = Mid(sInstruccionesOperandos(ejec1 - 1, 2), 3, 4)
      txtPuerto_P1.Visible = True
      imgArrow_P1.Visible = True
    End If
  End If
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
'Para Puerto P1 en WR
  If ejec1 > 0 And iEscrituraP1 Then
    If sInstruccionesOperandos(ejec1 - 1, 2) = "144" Or Mid(sInstruccionesOperandos(ejec1 - 1, 2), 1,
                                                  2) = "P1" Then
      txtPuerto_P1 = ""
      txtPuerto_P1.Visible = False
      imgArrow_P1.Visible = False
      txtPuerto_P1.ForeColor = QBColor(0)
      Form2.Puerto_P1WR
      Puerto_P1_color1
    End If
  End If
P2_Eprom_QBcolor1
P0_Latch_QBcolor1
Label4.ForeColor = QBColor(4)
Label4.Caption = "Direccionamiento a ROM"
Label4.Visible = True
txtDir_P2.Visible = True
txtDirDat_P0.Visible = True
imgArrow1.Visible = True
imgArrow2.Visible = True
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
'Envio del P2 y P0 como direcciones
txtDir_P2.Move 6120, 150
txtDirDat_P0.Move 5520, 1640
imgArrow1.Move 5760, 1680
imgArrow2.Move 5880, 160
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
'Segunda fase segundo estado
ALE_Latch_color1
PSEN_Eprom_QBcolor1
Latch_Eprom_QBcolor1
txtDir_P2.Move 9480, 150
txtDirDat_P0.Move 7320, 1640
imgArrow1.Move 7080, 1680
imgArrow2.Move 9240, 160
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
txtDir_P2.Move 9600, 630
txtDirDat_P0.Move 7800, 920
imgArrow1.Move 8040, 960
imgArrow2.Picture = frmImagenes.imgLeftarrow1.Picture
imgArrow2.Move 9360, 660
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
txtDir_P2.Visible = False
txtDirDat_P0.Visible = False
imgArrow1.Visible = False
imgArrow2.Visible = False
imgArrow2.Picture = frmImagenes.imgRightarrow1.Picture
frmDiagrama2.Refresh
'Primera fase tercer estado, segunda fase tercer estado,
'primera fase cuarto estado
```

```
ALE_Latch_color1
PSEN_Eprom_color1
P0_Latch_color1
Latch_Eprom_color1
Label4.ForeColor = QBColor(0)
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 4 * tiem: Loop
txtDirDat_P0.Move 7800, 2010
imgArrow1.Picture = frmImagenes.imgLeftarrow1.Picture
imgArrow1.Move 8040, 2040
Label5.ForeColor = QBColor(4)
Label5.Caption = "Lectura de instrucción"
Label5.Visible = True
txtDirDat_P0.Text = sInstruccionesOperandos(ejec1, 16)
txtDirDat_P0.Visible = True
imgArrow1.Visible = True
P0_Eprom_QBcolor1
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 0.7 * tiem: Loop  'Es 0.5 seg en realidad
txtDirDat_P0.Move 5640, 2660
imgArrow1.Move 5880, 2690
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 0.6 * tiem: Loop  'Es 0.5 seg en realidad
txtDirDat_P0.Move 2760, 2180
imgArrow1.Move 2520, 2220
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 0.7 * tiem: Loop  'Es 0.5 seg en realidad
Label4.Visible = False
Label5.Visible = False
txtDirDat_P0.Visible = False
imgArrow1.Visible = False
P0_Eprom_color1
frmDiagrama2.Refresh
txtDirDat_P0.Move 2760, 2180
txtDir_P2.Move 2760, 1190
imgArrow1.Picture = frmImagenes.imgRightarrow1.Picture
imgArrow2.Move 2520, 1200
'S4P2
ALE_Latch_QBcolor1
PSEN_Eprom_QBcolor1
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
P2_Eprom_color1
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
'S5P1
Label4.ForeColor = QBColor(4)
Label4.Caption = "Direccionamiento a RAM"
Label4.Visible = True
P2_Ram_QBcolor1
P0_Latch_QBcolor1
txtDir_P2.Text = sAddP2
txtDirDat_P0.Text = sAddP0
txtDir_P2.Visible = True
txtDirDat_P0.Visible = True
imgArrow1.Visible = True
imgArrow2.Visible = True
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
txtDir_P2.Move 7440, 150
txtDirDat_P0.Move 5520, 1640
imgArrow1.Move 5760, 1680
imgArrow2.Move 7200, 160
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
'S5P2
ALE_Latch_color1
Latch_Ram_QBcolor1
txtDir_P2.Move 9890, 1440
txtDirDat_P0.Move 7320, 1640
imgArrow1.Move 7080, 1680
```

```
imgArrow2.Move 9900, 1200
imgArrow2.Picture = frmImagenes.imgDownarrow1.Picture
imgArrow2.Height = 255
imgArrow2.Width = 150
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 3 * tiem: Loop
txtDir_P2.Move 9360, 3360
txtDirDat_P0.Move 7560, 3690
imgArrow1.Move 7800, 3720
imgArrow2.Move 9120, 3390
imgArrow2.Picture = frmImagenes.imgLeftarrow1.Picture
imgArrow2.Height = 150
imgArrow2.Width = 255
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 3 * tiem: Loop
'Ya estan incluidos arriba S6P1,S6P2,
txtDir_P2.Visible = False
txtDirDat_P0.Visible = False
imgArrow1.Visible = False
imgArrow2.Visible = False
txtDir_P2.Move 2760, 1190
txtDirDat_P0.Move 2760, 2180
imgArrow1.Move 2520, 2220
imgArrow2.Picture = frmImagenes.imgRightarrow1.Picture
imgArrow2.Move 2520, 1200
'CICLO 2:
'S1P1, S1P2
Label4.ForeColor = QBColor(0)
RD_NoAdd_QBcolor1
P0_Latch_color1
Latch_Ram_color1
txtDirDat_P0.Text = "00"    'sValor_InH_Buffer
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 2 * tiem: Loop
'Finaliza primer ciclo de maquina
Form2.Temp_Cont_0
Form2.Temp_Cont_1
Start = Timer: Do While Timer < Start + 4 * tiem: Loop
'CICLO 2 = S1P1 hasta S2P1
'CICLO 2
'S2P2, S3P1, S3P2
Label5.Caption = "Lectura de RAM"
Label5.Visible = True
P0_NoAdd_QBcolor1
txtDirDat_P0.Move 7060, 3740
imgArrow1.Move 7140, 3980
imgArrow1.Picture = frmImagenes.imgUparrow1.Picture
imgArrow1.Height = 255
imgArrow1.Width = 150
Label10.Caption = "de donde?"
Label10.Visible = True
txtDirDat_P0.Visible = True
imgArrow1.Visible = True
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 1.5 * tiem: Loop
txtDirDat_P0.Move 7060, 3240
imgArrow1.Move 7140, 3480
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 1.5 * tiem: Loop
txtDirDat_P0.Move 5640, 2660
imgArrow1.Move 5880, 2690
imgArrow1.Picture = frmImagenes.imgLeftarrow1.Picture
imgArrow1.Height = 150
imgArrow1.Width = 255
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 1.5 * tiem: Loop
txtDirDat_P0.Move 2760, 2180
imgArrow1.Move 2520, 2220
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 1.5 * tiem: Loop
Label4.Visible = False
```

```
Label5.Visible = False
'S4P1
txtDirDat_P0.Visible = False
imgArrow1.Visible = False
Label10.Visible = False
imgArrow1.Picture = frmImagenes.imgRightarrow1.Picture
RD_NoAdd_color1
P0_NoAdd_color1
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 2 * tiem: Loop
'S4P2
ALE_Latch_QBcolor1
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
P2_Ram_color1
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
'S5P1
txtDir_P2.Text = sInstruccionesOperandos(ejec1, 19)
txtDirDat_P0.Text = sInstruccionesOperandos(ejec1, 20)
txtDir_P2.Visible = True
txtDirDat_P0.Visible = True
imgArrow1.Visible = True
imgArrow2.Visible = True
Label4.ForeColor = QBColor(4)
Label4.Caption = "Direccionamiento a ROM"
Label4.Visible = True
P2_Eprom_QBcolor1
P0_Latch_QBcolor1
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
'Envio del P2 y P0 como direcciones
txtDir_P2.Move 6120, 150
txtDirDat_P0.Move 5520, 1640
imgArrow1.Move 5760, 1680
imgArrow2.Move 5880, 160
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
'Segunda fase segundo estado
ALE_Latch_color1
PSEN_Eprom_QBcolor1
Latch_Eprom_QBcolor1
txtDir_P2.Move 9480, 150
txtDirDat_P0.Move 7320, 1640
imgArrow1.Move 7080, 1680
imgArrow2.Move 9240, 160
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
txtDir_P2.Move 9600, 630
txtDirDat_P0.Move 7800, 920
imgArrow1.Move 8040, 960
imgArrow2.Picture = frmImagenes.imgLeftarrow1.Picture
imgArrow2.Move 9360, 660
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
txtDir_P2.Visible = False
txtDirDat_P0.Visible = False
imgArrow1.Visible = False
imgArrow2.Visible = False
imgArrow2.Picture = frmImagenes.imgRightarrow1.Picture
frmDiagrama2.Refresh
'Primera fase tercer estado, segunda fase tercer estado,
'primera fase cuarto estado
ALE_Latch_color1
PSEN_Eprom_color1
P0_Latch_color1
Latch_Eprom_color1
Label4.ForeColor = QBColor(0)
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 4 * tiem: Loop
Label5.ForeColor = QBColor(4)
```

```
Label5.Caption = "Lectura no válida"
Label5.Visible = True
P0_Eprom_QBcolor1
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 0.7 * tiem: Loop  'Es 0.5 seg en realidad
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 0.6 * tiem: Loop  'Es 0.5 seg en realidad
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 0.7 * tiem: Loop  'Es 0.5 seg en realidad
label1.Visible = False
Label2.Visible = False
Label3.Visible = False
Label4.Visible = False
Label5.Visible = False
'Finaliza segundo ciclo de maquina
Form2.Temp_Cont_0
Form2.Temp_Cont_1
'S1P2
txtDir_P2.Move 2760, 1190
txtDirDat_P0.Move 2760, 2180
imgArrow1.Move 2520, 2220
imgArrow2.Move 2520, 1200
frmDiagrama2.Refresh
Form2.Atencion_Interrupciones
End Sub

Sub Byte1_2CiclomovxW_1()
EscrituraP1
'Cada fase tendrá una duración de 2tiem segundo
tiem = Frecuencia_Oscilador
icolor = 16777215   'Line1.BorderColor
icolor2 = 16777215  'Shape1.BackColor
icolor3 = 8421504   'Shape4.BackColor
label1.Caption = saMatrizInstrucciones(ejec1)
Label2.Caption = "#Bytes = " & sInstruccionesOperandos(ejec1, 8)
Label3.Caption = "#C.Maq = " & sInstruccionesOperandos(ejec1, 9)
label1.Visible = True
Label2.Visible = True
Label3.Visible = True
txtDir_P2.Text = sInstruccionesOperandos(ejec1, 19)
txtDirDat_P0.Text = sInstruccionesOperandos(ejec1, 20)
'Segunda fase primer estado, primera fase segundo estado,
ALE_Latch_QBcolor1
PSEN_Eprom_QBcolor1
P2_Eprom_QBcolor1
P0_Eprom_color1
'Para Puerto P1 en WR
   If ejec1 > 0 And iEscrituraP1 Then
      If sInstruccionesOperandos(ejec1 - 1, 2) = "144" Or Mid(sInstruccionesOperandos(ejec1 - 1, 2), 1,
                                                  2) = "P1" Then
         Puerto_P1_QBcolor1
      End If
   End If
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
P2_Eprom_color1
'Para Puerto P1 en WR
   If ejec1 > 0 And iEscrituraP1 Then
      If sInstruccionesOperandos(ejec1 - 1, 2) = "144" Then
         txtPuerto_P1.Text = sMemoriaRAMintH(144)
         txtPuerto_P1.Visible = True
         imgArrow_P1.Visible = True
      ElseIf Mid(sInstruccionesOperandos(ejec1 - 1, 2), 1, 2) = "P1" Then
         txtPuerto_P1.ForeColor = QBColor(12)
         txtPuerto_P1.Text = Mid(sInstruccionesOperandos(ejec1 - 1, 2), 3, 4)
         txtPuerto_P1.Visible = True
         imgArrow_P1.Visible = True
      End If
   End If
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
```

```
'Para Puerto P1 en WR
  If ejec1 > 0 And iEscrituraP1 Then
    If sInstruccionesOperandos(ejec1 - 1, 2) = "144" Or Mid(sInstruccionesOperandos(ejec1 - 1, 2), 1,
                                                    2) = "P1" Then
      txtPuerto_P1 = ""
      txtPuerto_P1.Visible = False
      imgArrow_P1.Visible = False
      txtPuerto_P1.ForeColor = QBColor(0)
      Form2.Puerto_P1WR
      Puerto_P1_color1
    End If
  End If
P2_Eprom_QBcolor1
P0_Latch_QBcolor1
Label4.ForeColor = QBColor(4)
Label4.Caption = "Direccionamiento a ROM"
Label4.Visible = True
txtDir_P2.Move 2760, 1190
txtDir_P2.Visible = True
txtDirDat_P0.Visible = True
imgArrow1.Visible = True
imgArrow2.Visible = True
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
'Envio del P2 y P0 como dirrecciones
txtDir_P2.Move 6120, 150
txtDirDat_P0.Move 5520, 1640
imgArrow1.Move 5760, 1680
imgArrow2.Move 5880, 160
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
'Segunda fase segundo estado
ALE_Latch_color1
PSEN_Eprom_QBcolor1
Latch_Eprom_QBcolor1
txtDir_P2.Move 9480, 150
txtDirDat_P0.Move 7320, 1640
imgArrow1.Move 7080, 1680
imgArrow2.Move 9240, 160
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
txtDir_P2.Move 9600, 630
txtDirDat_P0.Move 7800, 920
imgArrow1.Move 8040, 960
imgArrow2.Picture = frmImagenes.imgLeftarrow1.Picture
imgArrow2.Move 9360, 660
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
txtDir_P2.Visible = False
txtDirDat_P0.Visible = False
imgArrow1.Visible = False
imgArrow2.Visible = False
imgArrow2.Picture = frmImagenes.imgRightarrow1.Picture
frmDiagrama2.Refresh
'Primera fase tercer estado, segunda fase tercer estado,
'primera fase cuarto estado
ALE_Latch_color1
PSEN_Eprom_color1
P0_Latch_color1
Latch_Eprom_color1
Label4.ForeColor = QBColor(0)
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 4 * tiem: Loop
txtDirDat_P0.Move 7800, 2010
imgArrow1.Picture = frmImagenes.imgLeftarrow1.Picture
imgArrow1.Move 8040, 2040
Label5.ForeColor = QBColor(4)
Label5.Caption = "Lectura de instrucción"
Label5.Visible = True
txtDirDat_P0.Text = sInstruccionesOperandos(ejec1, 5)
txtDirDat_P0.Visible = True
```

```
imgArrow1.Visible = True
P0_Eprom_QBcolor1
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 0.7 * tiem: Loop_P0.Move 5640, 2660
imgArrow1.Move 5880, 2690
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 0.6 * tiem: Loop
txtDirDat_P0.Move 2760, 2180
imgArrow1.Move 2520, 2220
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 0.7 * tiem: Loop
Label4.Visible = False
Label5.Visible = False
txtDirDat_P0.Visible = False
imgArrow1.Visible = False
P0_Eprom_color1
frmDiagrama2.Refresh
txtDir_P2.Move 2760, 1190
imgArrow1.Picture = frmImagenes.imgRightarrow1.Picture
imgArrow2.Move 2520, 1200
'S4P2
ALE_Latch_QBcolor1
PSEN_Eprom_QBcolor1
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
P2_Eprom_color1
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
'S5P1
Label4.ForeColor = QBColor(4)
Label4.Caption = "Direccionamiento a RAM"
Label4.Visible = True
P2_Ram_QBcolor1
P0_Latch_QBcolor1
txtDir_P2.Text = sAddP2
txtDirDat_P0.Text = sAddP0
txtDir_P2.Visible = True
txtDirDat_P0.Visible = True
imgArrow1.Visible = True
imgArrow2.Visible = True
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
txtDir_P2.Move 7440, 160
txtDirDat_P0.Move 5520, 1640
imgArrow1.Move 5760, 1680
imgArrow2.Move 7200, 160
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
'S5P2
ALE_Latch_color1
Latch_Ram_QBcolor1
txtDir_P2.Move 9890, 1440
txtDirDat_P0.Move 7320, 1640
imgArrow1.Move 7080, 1680
imgArrow2.Picture = frmImagenes.imgDownarrow1.Picture
imgArrow2.Move 9900, 1200
imgArrow2.Height = 255
imgArrow2.Width = 150
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 1.5 * tiem: Loop
txtDir_P2.Move 9360, 3360
txtDirDat_P0.Move 7560, 3690
imgArrow1.Move 7800, 3720
imgArrow2.Picture = frmImagenes.imgLeftarrow1.Picture
imgArrow2.Move 9120, 3390
imgArrow2.Height = 150
imgArrow2.Width = 255
frmDiagrama2.Refresh
'S6P1 solo media transicionde reloj
Start = Timer: Do While Timer < Start + 1.5 * tiem: Loop
Label4.ForeColor = QBColor(0)
```

```
'S6P1 la otra mitad de la transicion,
txtDir_P2.Visible = False
txtDirDat_P0.Visible = False
imgArrow1.Visible = False
imgArrow2.Visible = False
txtDir_P2.Move 2760, 1190
txtDirDat_P0.Move 2760, 2180
imgArrow1.Move 2520, 2220
imgArrow2.Picture = frmImagenes.imgRightarrow1.Picture
imgArrow2.Move 2520, 1200
P0_Latch_color1
Latch_Ram_color1
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
'S6P2
Label5.Caption = "Escritura en RAM"
Label5.Visible = True
P0_Ram_QBcolor1
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 2 * tiem: Loop
'CICLO 2 = S1P1 hasta S3P2
WR_Ram_QBcolor1
txtDirDat_P0.Text = sDatP0
txtDirDat_P0.Move 2760, 2180
txtDirDat_P0.Visible = True
imgArrow1.Visible = True
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 2 * tiem: Loop
'Finaliza primer ciclo de máquina
Form2.Temp_Cont_0
Form2.Temp_Cont_1
Start = Timer: Do While Timer < Start + 1 * tiem: Loop
txtDirDat_P0.Move 5880, 2660
imgArrow1.Move 5640, 2690
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 3 * tiem: Loop
txtDirDat_P0.Move 7060, 3480
imgArrow1.Picture = frmImagenes.imgDownarrow1.Picture
imgArrow1.Move 7140, 3240
imgArrow1.Height = 255
imgArrow1.Width = 150
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 3 * tiem: Loop
txtDirDat_P0.Move 7560, 4770
imgArrow1.Picture = frmImagenes.imgRightarrow1.Picture
imgArrow1.Move 7800, 4800
imgArrow1.Height = 150
imgArrow1.Width = 255
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 3 * tiem: Loop
'S4P1
WR_Ram_color1
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 2 * tiem: Loop
'S4P2 medio ciclo
ALE_Latch_QBcolor1
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
Label4.Visible = False
Label5.Visible = False
'S4P2 medio siguiente
txtDirDat_P0.Visible = False
imgArrow1.Visible = False
txtDirDat_P0.Move 2760, 2180
imgArrow1.Move 2520, 2220
P0_Ram_color1
P2_Ram_color1
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
'S5P1
txtDir_P2.Text = sInstruccionesOperandos(ejec1, 19)
```

```
txtDirDat_P0.Text = sInstruccionesOperandos(ejec1, 20)
txtDir_P2.Visible = True
txtDirDat_P0.Visible = True
imgArrow1.Visible = True
imgArrow2.Visible = True
Label4.ForeColor = QBColor(4)
Label4.Caption = "Direccionamiento a ROM"
Label4.Visible = True
P2_Eprom_QBcolor1
P0_Latch_QBcolor1
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
'Envio del P2 y P0 como direcciones
txtDir_P2.Move 6120, 150
txtDirDat_P0.Move 5520, 1640
imgArrow1.Move 5760, 1680
imgArrow2.Move 5880, 160
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
'Segunda fase segundo estado
ALE_Latch_color1
PSEN_Eprom_QBcolor1
Latch_Eprom_QBcolor1
txtDir_P2.Move 9480, 150
txtDirDat_P0.Move 7320, 1640
imgArrow1.Move 7080, 1680
imgArrow2.Move 9240, 160
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
txtDir_P2.Move 9600, 630
txtDirDat_P0.Move 7800, 920
imgArrow1.Move 8040, 960
imgArrow2.Picture = frmImagenes.imgLeftarrow1.Picture
imgArrow2.Move 9360, 660
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
txtDir_P2.Visible = False
txtDirDat_P0.Visible = False
imgArrow1.Visible = False
imgArrow2.Visible = False
imgArrow2.Picture = frmImagenes.imgRightarrow1.Picture
frmDiagrama2.Refresh
'Primera fase tercer estado, segunda fase tercer estado,
'primera fase cuarto estado
ALE_Latch_color1
PSEN_Eprom_color1
P0_Latch_color1
Latch_Eprom_color1
Label4.ForeColor = QBColor(0)
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 4 * tiem: Loop
Label5.ForeColor = QBColor(4)
Label5.Caption = "Lectura no válida"
Label5.Visible = True
P0_Eprom_QBcolor1
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 0.7 * tiem: Loop  'Es 0.5 seg en realidad
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 0.6 * tiem: Loop  'Es 0.5 seg en realidad
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 0.7 * tiem: Loop  'Es 0.5 seg en realidad
label1.Visible = False
Label2.Visible = False
Label3.Visible = False
Label4.Visible = False
Label5.Visible = False
'Finaliza segundo ciclo de máquina
Form2.Temp_Cont_0
Form2.Temp_Cont_1
'S1P2
txtDir_P2.Move 2760, 1190
```

```
txtDirDat_P0.Move 2760, 2180
imgArrow1.Move 2520, 2220
imgArrow2.Move 2520, 1200
frmDiagrama2.Refresh
Form2.Atencion_Interrupciones
End Sub

Sub Byte1_2CiclomovxW_Latch()
EscrituraP1
'Cada fase tendrá una duración de 2tiem segundo
tiem = Frecuencia_Oscilador
icolor = 16777215   'Line1.BorderColor
icolor2 = 16777215  'Shape1.BackColor
icolor3 = 8421504   'Shape4.BackColor
label1.Caption = saMatrizInstrucciones(ejec1)
Label2.Caption = "#Bytes = " & sInstruccionesOperandos(ejec1, 8)
Label3.Caption = "#C.Maq = " & sInstruccionesOperandos(ejec1, 9)
label1.Visible = True
Label2.Visible = True
Label3.Visible = True
txtDir_P2.Text = sInstruccionesOperandos(ejec1, 19)
txtDirDat_P0.Text = sInstruccionesOperandos(ejec1, 20)
'Segunda fase primer estado, primera fase segundo estado,
ALE_Latch_QBcolor1
PSEN_Eprom_QBcolor1
P2_Eprom_QBcolor1
P0_Eprom_color1
'Para Puerto P1 en WR
  If ejec1 > 0 And iEscrituraP1 Then
    If sInstruccionesOperandos(ejec1 - 1, 2) = "144" Or Mid(sInstruccionesOperandos(ejec1 - 1, 2), 1,
                                                2) = "P1" Then
      Puerto_P1_QBcolor1
    End If
  End If
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
P2_Eprom_color1
'Para Puerto P1 en WR
  If ejec1 > 0 And iEscrituraP1 Then
    If sInstruccionesOperandos(ejec1 - 1, 2) = "144" Then
      txtPuerto_P1.Text = sMemoriaRAMintH(144)
      txtPuerto_P1.Visible = True
      imgArrow_P1.Visible = True
    ElseIf Mid(sInstruccionesOperandos(ejec1 - 1, 2), 1, 2) = "P1" Then
      txtPuerto_P1.ForeColor = QBColor(12)
      txtPuerto_P1.Text = Mid(sInstruccionesOperandos(ejec1 - 1, 2), 3, 4)
      txtPuerto_P1.Visible = True
      imgArrow_P1.Visible = True
    End If
  End If
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
'Para Puerto P1 en WR
  If ejec1 > 0 And iEscrituraP1 Then
    If sInstruccionesOperandos(ejec1 - 1, 2) = "144" Or Mid(sInstruccionesOperandos(ejec1 - 1, 2), 1,
                                                2) = "P1" Then
      txtPuerto_P1 = ""
      txtPuerto_P1.Visible = False
      imgArrow_P1.Visible = False
      txtPuerto_P1.ForeColor = QBColor(0)
      Form2.Puerto_P1WR
      Puerto_P1_color1
    End If
  End If
P2_Eprom_QBcolor1
P0_Latch_QBcolor1
Label4.ForeColor = QBColor(4)
Label4.Caption = "Direccionamiento a ROM"
Label4.Visible = True
txtDir_P2.Move 2760, 1190
txtDir_P2.Visible = True
```

```
txtDirDat_P0.Visible = True
imgArrow1.Visible = True
imgArrow2.Visible = True
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
'Envio del P2 y P0 como dirrecciones
txtDir_P2.Move 6120, 150
txtDirDat_P0.Move 5520, 1640
imgArrow1.Move 5760, 1680
imgArrow2.Move 5880, 160
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
'Segunda fase segundo estado
ALE_Latch_color1
PSEN_Eprom_QBcolor1
Latch_Eprom_QBcolor1
txtDir_P2.Move 9480, 150
txtDirDat_P0.Move 7320, 1640
imgArrow1.Move 7080, 1680
imgArrow2.Move 9240, 160
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
txtDir_P2.Move 9600, 630
txtDirDat_P0.Move 7800, 920
imgArrow1.Move 8040, 960
imgArrow2.Picture = frmImagenes.imgLeftarrow1.Picture
imgArrow2.Move 9360, 660
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
txtDir_P2.Visible = False
txtDirDat_P0.Visible = False
imgArrow1.Visible = False
imgArrow2.Visible = False
imgArrow2.Picture = frmImagenes.imgRightarrow1.Picture
frmDiagrama2.Refresh
'Primera fase tercer estado, segunda fase tercer estado,
'primera fase cuarto estado
ALE_Latch_color1
PSEN_Eprom_color1
P0_Latch_color1
Latch_Eprom_color1
Label4.ForeColor = QBColor(0)
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 4 * tiem: Loop
txtDirDat_P0.Move 7800, 2010
imgArrow1.Picture = frmImagenes.imgLeftarrow1.Picture
imgArrow1.Move 8040, 2040
Label5.ForeColor = QBColor(4)
Label5.Caption = "Lectura de instrucción"
Label5.Visible = True
txtDirDat_P0.Text = sInstruccionesOperandos(ejec1, 5)
txtDirDat_P0.Visible = True
imgArrow1.Visible = True
P0_Eprom_QBcolor1
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 0.7 * tiem: Loop_P0.Move 5640, 2660
imgArrow1.Move 5880, 2690
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 0.6 * tiem: Loop
txtDirDat_P0.Move 2760, 2180
imgArrow1.Move 2520, 2220
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 0.7 * tiem: Loop
Label4.Visible = False
Label5.Visible = False
txtDirDat_P0.Visible = False
imgArrow1.Visible = False
P0_Eprom_color1
frmDiagrama2.Refresh
txtDir_P2.Move 2760, 1190
imgArrow1.Picture = frmImagenes.imgRightarrow1.Picture
```

```
imgArrow2.Move 2520, 1200
'S4P2
ALE_Latch_QBcolor1
PSEN_Eprom_QBcolor1
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
P2_Eprom_color1
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
'S5P1
Label4.ForeColor = QBColor(4)
Label4.Caption = "Direccionamiento a RAM"
Label4.Visible = True
P2_Ram_QBcolor1
P0_Latch_QBcolor1
txtDir_P2.Text = sAddP2
txtDirDat_P0.Text = sAddP0
txtDir_P2.Visible = True
txtDirDat_P0.Visible = True
imgArrow1.Visible = True
imgArrow2.Visible = True
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
txtDir_P2.Move 7440, 160
txtDirDat_P0.Move 5520, 1640
imgArrow1.Move 5760, 1680
imgArrow2.Move 7200, 160
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
'S5P2
ALE_Latch_color1
Latch_Ram_QBcolor1
txtDir_P2.Move 9890, 1440
txtDirDat_P0.Move 7320, 1640
imgArrow1.Move 7080, 1680
imgArrow2.Picture = frmImagenes.imgDownarrow1.Picture
imgArrow2.Move 9900, 1200
imgArrow2.Height = 255
imgArrow2.Width = 150
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 1.5 * tiem: Loop
txtDir_P2.Move 9360, 3360
txtDirDat_P0.Move 7560, 3690
imgArrow1.Move 7800, 3720
imgArrow2.Picture = frmImagenes.imgLeftarrow1.Picture
imgArrow2.Move 9120, 3390
imgArrow2.Height = 150
imgArrow2.Width = 255
frmDiagrama2.Refresh
'S6P1 solo media transicionde reloj
Start = Timer: Do While Timer < Start + 1.5 * tiem: Loop
Label4.ForeColor = QBColor(0)
'S6P1 la otra mitad de la transicion,
txtDir_P2.Visible = False
txtDirDat_P0.Visible = False
imgArrow1.Visible = False
imgArrow2.Visible = False
txtDir_P2.Move 2760, 1190
txtDirDat_P0.Move 2760, 2180
imgArrow1.Move 2520, 2220
imgArrow2.Picture = frmImagenes.imgRightarrow1.Picture
imgArrow2.Move 2520, 1200
P0_Latch_color1
Latch_Ram_color1
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
'S6P2
Label5.Caption = "Escritura en RAM"
Label5.Visible = True
P0_Latch377_QBcolor1
frmDiagrama2.Refresh
```

```
Start = Timer: Do While Timer < Start + 2 * tiem: Loop
'CICLO 2 = S1P1 hasta S3P2
WR_Latch377_QBcolor1
txtDirDat_P0.Text = sDatP0
txtDirDat_P0.Move 2760, 2180
txtDirDat_P0.Visible = True
imgArrow1.Visible = True
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 2 * tiem: Loop
'Finaliza primer ciclo de máquina
Form2.Temp_Cont_0
Form2.Temp_Cont_1
Start = Timer: Do While Timer < Start + 1 * tiem: Loop
txtDirDat_P0.Move 5880, 2660
imgArrow1.Move 5640, 2690
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 3 * tiem: Loop
txtDirDat_P0.Move 7060, 3480
imgArrow1.Picture = frmImagenes.imgDownarrow1.Picture
imgArrow1.Move 7140, 3240
imgArrow1.Height = 255
imgArrow1.Width = 150
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 3 * tiem: Loop
txtDirDat_P0.Move 7020, 5190
imgArrow1.Picture = frmImagenes.imgLeftarrow1.Picture
imgArrow1.Move 6750, 5220
imgArrow1.Height = 150
imgArrow1.Width = 255
'Salida de datos del Latch a los LEDs
For led1 = 0 To 7
   If iMemoriaRAMint(224, 7 - led1) = 1 Then
      LED(led1).BackColor = QBColor(12)
      LED(led1).BorderColor = QBColor(12)
   Else
      LED(led1).BackColor = icolor
      LED(led1).BorderColor = icolor
   End If
Next led1
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 3 * tiem: Loop
'S4P1
WR_Latch377_color1
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 2 * tiem: Loop
'S4P2 medio ciclo
ALE_Latch_QBcolor1
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
Label4.Visible = False
Label5.Visible = False
'S4P2 medio siguiente
txtDirDat_P0.Visible = False
imgArrow1.Visible = False
txtDirDat_P0.Move 2760, 2180
imgArrow1.Move 2520, 2220
imgArrow1.Picture = frmImagenes.imgRightarrow1.Picture
P0_Latch377_color1
P2_Ram_color1
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
'S5P1
txtDir_P2.Text = sInstruccionesOperandos(ejec1, 19)
txtDirDat_P0.Text = sInstruccionesOperandos(ejec1, 20)
txtDir_P2.Visible = True
txtDirDat_P0.Visible = True
imgArrow1.Visible = True
imgArrow2.Visible = True
Label4.ForeColor = QBColor(4)
Label4.Caption = "Direccionamiento a ROM"
Label4.Visible = True
```

```
P2_Eprom_QBcolor1
P0_Latch_QBcolor1
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
'Envio del P2 y P0 como direcciones
txtDir_P2.Move 6120, 150
txtDirDat_P0.Move 5520, 1640
imgArrow1.Move 5760, 1680
imgArrow2.Move 5880, 160
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
'Segunda fase segundo estado
ALE_Latch_color1
PSEN_Eprom_QBcolor1
Latch_Eprom_QBcolor1
txtDir_P2.Move 9480, 150
txtDirDat_P0.Move 7320, 1640
imgArrow1.Move 7080, 1680
imgArrow2.Move 9240, 160
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
txtDir_P2.Move 9600, 630
txtDirDat_P0.Move 7800, 920
imgArrow1.Move 8040, 960
imgArrow2.Picture = frmImagenes.imgLeftarrow1.Picture
imgArrow2.Move 9360, 660
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
txtDir_P2.Visible = False
txtDirDat_P0.Visible = False
imgArrow1.Visible = False
imgArrow2.Visible = False
imgArrow2.Picture = frmImagenes.imgRightarrow1.Picture
frmDiagrama2.Refresh
'Primera fase tercer estado, segunda fase tercer estado,
'primera fase cuarto estado
ALE_Latch_color1
PSEN_Eprom_color1
P0_Latch_color1
Latch_Eprom_color1
Label4.ForeColor = QBColor(0)
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 4 * tiem: Loop
Label5.ForeColor = QBColor(4)
Label5.Caption = "Lectura no válida"
Label5.Visible = True
P0_Eprom_QBcolor1
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 0.7 * tiem: Loop  'Es 0.5 seg en realidad
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 0.6 * tiem: Loop  'Es 0.5 seg en realidad
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 0.7 * tiem: Loop  'Es 0.5 seg en realidad
label1.Visible = False
Label2.Visible = False
Label3.Visible = False
Label4.Visible = False
Label5.Visible = False
'Finaliza segundo ciclo de máquina
Form2.Temp_Cont_0
Form2.Temp_Cont_1
'S1P2
txtDir_P2.Move 2760, 1190
txtDirDat_P0.Move 2760, 2180
imgArrow1.Move 2520, 2220
imgArrow2.Move 2520, 1200
frmDiagrama2.Refresh
Form2.Atencion_Interrupciones
End Sub
```

Sub Byte1_2CiclomovxW_NoAdd()

```
EscrituraP1
'Cada fase tendrá una duración de 2tiem segundo
tiem = Frecuencia_Oscilador
icolor = 16777215  'Line1.BorderColor
icolor2 = 16777215  'Shape1.BackColor
icolor3 = 8421504  'Shape4.BackColor
label1.Caption = saMatrizInstrucciones(ejec1)
Label2.Caption = "#Bytes = " & sInstruccionesOperandos(ejec1, 8)
Label3.Caption = "#C.Maq = " & sInstruccionesOperandos(ejec1, 9)
label1.Visible = True
Label2.Visible = True
Label3.Visible = True
txtDir_P2.Text = sInstruccionesOperandos(ejec1, 19)
txtDirDat_P0.Text = sInstruccionesOperandos(ejec1, 20)
'Segunda fase primer estado, primera fase segundo estado,
ALE_Latch_QBcolor1
PSEN_Eprom_QBcolor1
P2_Eprom_QBcolor1
P0_Eprom_color1
'Para Puerto P1 en WR
  If ejec1 > 0 And iEscrituraP1 Then
    If sInstruccionesOperandos(ejec1 - 1, 2) = "144" Or Mid(sInstruccionesOperandos(ejec1 - 1, 2), 1,
                                                2) = "P1" Then
      Puerto_P1_QBcolor1
    End If
  End If
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
P2_Eprom_color1
'Para Puerto P1 en WR
  If ejec1 > 0 And iEscrituraP1 Then
    If sInstruccionesOperandos(ejec1 - 1, 2) = "144" Then
      txtPuerto_P1.Text = sMemoriaRAMintH(144)
      txtPuerto_P1.Visible = True
      imgArrow_P1.Visible = True
    ElseIf Mid(sInstruccionesOperandos(ejec1 - 1, 2), 1, 2) = "P1" Then
      txtPuerto_P1.ForeColor = QBColor(12)
      txtPuerto_P1.Text = Mid(sInstruccionesOperandos(ejec1 - 1, 2), 3, 4)
      txtPuerto_P1.Visible = True
      imgArrow_P1.Visible = True
    End If
  End If
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
'Para Puerto P1 en WR
  If ejec1 > 0 And iEscrituraP1 Then
    If sInstruccionesOperandos(ejec1 - 1, 2) = "144" Or Mid(sInstruccionesOperandos(ejec1 - 1, 2), 1,
                                                2) = "P1" Then
      txtPuerto_P1 = ""
      txtPuerto_P1.Visible = False
      imgArrow_P1.Visible = False
      txtPuerto_P1.ForeColor = QBColor(0)
      Form2.Puerto_P1WR
      Puerto_P1_color1
    End If
  End If
P2_Eprom_QBcolor1
P0_Latch_QBcolor1
Label4.ForeColor = QBColor(4)
Label4.Caption = "Direccionamiento a ROM"
Label4.Visible = True
txtDir_P2.Move 2760, 1190
txtDir_P2.Visible = True
txtDirDat_P0.Visible = True
imgArrow1.Visible = True
imgArrow2.Visible = True
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
'Envio del P2 y P0 como dirrecciones
txtDir_P2.Move 6120, 150
txtDirDat_P0.Move 5520, 1640
```

```
imgArrow1.Move 5760, 1680
imgArrow2.Move 5880, 160
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
'Segunda fase segundo estado
ALE_Latch_color1
PSEN_Eprom_QBcolor1
Latch_Eprom_QBcolor1
txtDir_P2.Move 9480, 150
txtDirDat_P0.Move 7320, 1640
imgArrow1.Move 7080, 1680
imgArrow2.Move 9240, 160
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
txtDir_P2.Move 9600, 630
txtDirDat_P0.Move 7800, 920
imgArrow1.Move 8040, 960
imgArrow2.Picture = frmImagenes.imgLeftarrow1.Picture
imgArrow2.Move 9360, 660
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
txtDir_P2.Visible = False
txtDirDat_P0.Visible = False
imgArrow1.Visible = False
imgArrow2.Visible = False
imgArrow2.Picture = frmImagenes.imgRightarrow1.Picture
frmDiagrama2.Refresh
'Primera fase tercer estado, segunda fase tercer estado,
'primera fase cuarto estado
ALE_Latch_color1
PSEN_Eprom_color1
P0_Latch_color1
Latch_Eprom_color1
Label4.ForeColor = QBColor(0)
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 4 * tiem: Loop
txtDirDat_P0.Move 7800, 2010
imgArrow1.Picture = frmImagenes.imgLeftarrow1.Picture
imgArrow1.Move 8040, 2040
Label5.ForeColor = QBColor(4)
Label5.Caption = "Lectura de instrucción"
Label5.Visible = True
txtDirDat_P0.Text = sInstruccionesOperandos(ejec1, 5)
txtDirDat_P0.Visible = True
imgArrow1.Visible = True
P0_Eprom_QBcolor1
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 0.7 * tiem: Loop_P0.Move 5640, 2660
imgArrow1.Move 5880, 2690
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 0.6 * tiem: Loop
txtDirDat_P0.Move 2760, 2180
imgArrow1.Move 2520, 2220
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 0.7 * tiem: Loop
Label4.Visible = False
Label5.Visible = False
txtDirDat_P0.Visible = False
imgArrow1.Visible = False
P0_Eprom_color1
frmDiagrama2.Refresh
txtDir_P2.Move 2760, 1190
imgArrow1.Picture = frmImagenes.imgRightarrow1.Picture
imgArrow2.Move 2520, 1200
'S4P2
ALE_Latch_QBcolor1
PSEN_Eprom_QBcolor1
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
P2_Eprom_color1
frmDiagrama2.Refresh
```

```
Start = Timer: Do While Timer < Start + tiem: Loop
'S5P1
Label4.ForeColor = QBColor(4)
Label4.Caption = "Direccionamiento a RAM"
Label4.Visible = True
P2_Ram_QBcolor1
P0_Latch_QBcolor1
txtDir_P2.Text = sAddP2
txtDirDat_P0.Text = sAddP0
txtDir_P2.Visible = True
txtDirDat_P0.Visible = True
imgArrow1.Visible = True
imgArrow2.Visible = True
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
txtDir_P2.Move 7440, 160
txtDirDat_P0.Move 5520, 1640
imgArrow1.Move 5760, 1680
imgArrow2.Move 7200, 160
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
'S5P2
ALE_Latch_color1
Latch_Ram_QBcolor1
txtDir_P2.Move 9890, 1440
txtDirDat_P0.Move 7320, 1640
imgArrow1.Move 7080, 1680
imgArrow2.Picture = frmImagenes.imgDownarrow1.Picture
imgArrow2.Move 9900, 1200
imgArrow2.Height = 255
imgArrow2.Width = 150
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 1.5 * tiem: Loop
txtDir_P2.Move 9360, 3360
txtDirDat_P0.Move 7560, 3690
imgArrow1.Move 7800, 3720
imgArrow2.Picture = frmImagenes.imgLeftarrow1.Picture
imgArrow2.Move 9120, 3390
imgArrow2.Height = 150
imgArrow2.Width = 255
frmDiagrama2.Refresh
'S6P1 solo media transicionde reloj
Start = Timer: Do While Timer < Start + 1.5 * tiem: Loop
Label4.ForeColor = QBColor(0)
'S6P1 la otra mitad de la transicion,
txtDir_P2.Visible = False
txtDirDat_P0.Visible = False
imgArrow1.Visible = False
imgArrow2.Visible = False
txtDir_P2.Move 2760, 1190
txtDirDat_P0.Move 2760, 2180
imgArrow1.Move 2520, 2220
imgArrow2.Picture = frmImagenes.imgRightarrow1.Picture
imgArrow2.Move 2520, 1200
P0_Latch_color1
Latch_Ram_color1
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
'S6P2
Label5.Caption = "Escritura en RAM"
Label5.Visible = True
P0_NoAdd_QBcolor1
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 2 * tiem: Loop
'CICLO 2 = S1P1 hasta S3P2
WR_NoAdd_QBcolor1
tx.DirDat_P0.Text = sDatP0
txtDirDat_P0.Move 2760, 2180
Label10.Caption = "a donde?"
Label10.Visible = True
txtDirDat_P0.Visible = True
```

```
imgArrow1.Visible = True
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 2 * tiem: Loop
'Finaliza primer ciclo de máquina
Form2.Temp_Cont_0
Form2.Temp_Cont_1
Start = Timer: Do While Timer < Start + 1 * tiem: Loop
txtDirDat_P0.Move 5880, 2660
imgArrow1.Move 5640, 2690
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 3 * tiem: Loop
txtDirDat_P0.Move 7060, 3480
imgArrow1.Picture = frmImagenes.imgDownarrow1.Picture
imgArrow1.Move 7140, 3240
imgArrow1.Height = 255
imgArrow1.Width = 150
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 3 * tiem: Loop
txtDirDat_P0.Move 7060, 3980
imgArrow1.Move 7140, 3740
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 3 * tiem: Loop
'S4P1
WR_NoAdd_color1
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 2 * tiem: Loop
'S4P2 medio ciclo
ALE_Latch_QBcolor1
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
Label4.Visible = False
Label5.Visible = False
'S4P2 medio siguiente
txtDirDat_P0.Visible = False
imgArrow1.Visible = False
Label10.Visible = False
txtDirDat_P0.Move 2760, 2180
imgArrow1.Move 2520, 2220
P0_NoAdd_color1
P2_Ram_color1
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
'S5P1
txtDir_P2.Text = sInstruccionesOperandos(ejec1, 19)
txtDirDat_P0.Text = sInstruccionesOperandos(ejec1, 20)
imgArrow1.Picture = frmImagenes.imgRightarrow1.Picture
imgArrow1.Height = 150
imgArrow1.Width = 255
txtDir_P2.Visible = True
txtDirDat_P0.Visible = True
imgArrow1.Visible = True
imgArrow2.Visible = True
Label4.ForeColor = QBColor(4)
Label4.Caption = "Direccionamiento a ROM"
Label4.Visible = True
P2_Eprom_QBcolor1
P0_Latch_QBcolor1
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
'Envio del P2 y P0 como direcciones
txtDir_P2.Move 6120, 150
txtDirDat_P0.Move 5520, 1640
imgArrow1.Move 5760, 1680
imgArrow2.Move 5880, 160
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
'Segunda fase segundo estado
ALE_Latch_color1
PSEN_Eprom_QBcolor1
Latch_Eprom_QBcolor1
txtDir_P2.Move 9480, 150
```

```
txtDirDat_P0.Move 7320, 1640
imgArrow1.Move 7080, 1680
imgArrow2.Move 9240, 160
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
txtDir_P2.Move 9600, 630
txtDirDat_P0.Move 7800, 920
imgArrow1.Move 8040, 960
imgArrow2.Picture = frmImagenes.imgLeftarrow1.Picture
imgArrow2.Move 9360, 660
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
txtDir_P2.Visible = False
txtDirDat_P0.Visible = False
imgArrow1.Visible = False
imgArrow2.Visible = False
imgArrow2.Picture = frmImagenes.imgRightarrow1.Picture
frmDiagrama2.Refresh
'Primera fase tercer estado, segunda fase tercer estado,
'primera fase cuarto estado
ALE_Latch_color1
PSEN_Eprom_color1
P0_Latch_color1
Latch_Eprom_color1
Label4.ForeColor = QBColor(0)
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 4 * tiem: Loop
Label5.ForeColor = QBColor(4)
Label5.Caption = "Lectura no válida"
Label5.Visible = True
P0_Eprom_QBcolor1
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 0.7 * tiem: Loop  'Es 0.5 seg en realidad
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 0.6 * tiem: Loop  'Es 0.5 seg en realidad
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 0.7 * tiem: Loop  'Es 0.5 seg en realidad
label1.Visible = False
Label2.Visible = False
Label3.Visible = False
Label4.Visible = False
Label5.Visible = False
'Finaliza segundo ciclo de máquina
Form2.Temp_Cont_0
Form2.Temp_Cont_1
'S1P2
txtDir_P2.Move 2760, 1190
txtDirDat_P0.Move 2760, 2180
imgArrow1.Move 2520, 2220
imgArrow2.Move 2520, 1200
frmDiagrama2.Refresh
Form2.Atencion_Interrupciones
End Sub

Sub Byte1_4Ciclo_1()
EscrituraP1
'Cada fase tendrá una duración de 2tiem segundo
tiem = Frecuencia_Oscilador
icolor = 16777215   'Line1.BorderColor
icolor2 = 16777215  'Shape1.BackColor
icolor3 = 8421504   'Shape4.BackColor
label1.Caption = saMatrizInstrucciones(ejec1)
Label2.Caption = "#Bytes = " & sInstruccionesOperandos(ejec1, 8)
Label3.Caption = "#C.Maq = " & sInstruccionesOperandos(ejec1, 9)
label1.Visible = True
Label2.Visible = True
Label3.Visible = True
For icolor1 = 1 To 8
txtDir_P2.Text = sInstruccionesOperandos(ejec1, 19)
txtDirDat_P0.Text = sInstruccionesOperandos(ejec1, 20)
'Segunda fase primer estado, primera fase segundo estado,
```

```
ALE_Latch_QBcolor1
PSEN_Eprom_QBcolor1
P2_Eprom_QBcolor1
P0_Eprom_color1
'Para Puerto P1 en WR
   If icolor1 = 1 And ejec1 > 0 And iEscrituraP1 Then
      If sInstruccionesOperandos(ejec1 - 1, 2) = "144" Or Mid(sInstruccionesOperandos(ejec1 - 1, 2), 1,
                                                    2) = "P1" Then
         Puerto_P1_QBcolor1
      End If
   End If
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
P2_Eprom_color1
'Para Puerto P1 en WR
   If icolor1 = 1 And ejec1 > 0 And iEscrituraP1 Then
      If sInstruccionesOperandos(ejec1 - 1, 2) = "144" Then
         txtPuerto_P1.Text = sMemoriaRAMintH(144)
         txtPuerto_P1.Visible = True
         imgArrow_P1.Visible = True
      ElseIf Mid(sInstruccionesOperandos(ejec1 - 1, 2), 1, 2) = "P1" Then
         txtPuerto_P1.ForeColor = QBColor(12)
         txtPuerto_P1.Text = Mid(sInstruccionesOperandos(ejec1 - 1, 2), 3, 4)
         txtPuerto_P1.Visible = True
         imgArrow_P1.Visible = True
      End If
   End If
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
'Para Puerto P1 en WR
   If icolor1 = 1 And ejec1 > 0 And iEscrituraP1 Then
      If sInstruccionesOperandos(ejec1 - 1, 2) = "144" Or Mid(sInstruccionesOperandos(ejec1 - 1, 2), 1,
                                                    2) = "P1" Then
         txtPuerto_P1 = ""
         txtPuerto_P1.Visible = False
         imgArrow_P1.Visible = False
         txtPuerto_P1.ForeColor = QBColor(0)
         Form2.Puerto_P1WR
         Puerto_P1_color1
      End If
   End If
P2_Eprom_QBcolor1_Latch_QBcolor1
Label4.Caption = "Direccionamiento a ROM"
Label4.Visible = True
Label4.ForeColor = QBColor(4)
txtDir_P2.Visible = True
txtDirDat_P0.Visible = True
imgArrow1.Visible = True
imgArrow2.Visible = True
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
'Envio del P2 y P0 como dirrecciones
txtDir_P2.Move 6120, 150
txtDirDat_P0.Move 5520, 1640
imgArrow1.Move 5760, 1680
imgArrow2.Move 5880, 160
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
'Segunda fase segundo estado
ALE_Latch_color1
PSEN_Eprom_QBcolor1
Latch_Eprom_QBcolor1
txtDir_P2.Move 9480, 150
txtDirDat_P0.Move 7320, 1640
imgArrow1.Move 7080, 1680
imgArrow2.Move 9240, 160
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
txtDir_P2.Move 9600, 630
txtDirDat_P0.Move 7800, 920
imgArrow1.Move 8040, 960
```

```
imgArrow2.Picture = frmImagenes.imgLeftarrow1.Picture
imgArrow2.Move 9360, 660
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
txtDir_P2.Visible = False
txtDirDat_P0.Visible = False
imgArrow1.Visible = False
imgArrow2.Visible = False
imgArrow2.Picture = frmImagenes.imgRightarrow1.Picture
imgArrow2.Move 2520, 1200
frmDiagrama2.Refresh
'Primera fase tercer estado, segunda fase tercer estado,
'primera fase cuarto estado
ALE_Latch_color1
PSEN_Eprom_color1
P0_Latch_color1
Latch_Eprom_color1
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 4 * tiem: Loop
txtDirDat_P0.Move 7800, 2010
txtDirDat_P0.Text = sInstruccionesOperandos(ejec1, 16)
If icolor1 = 1 Then
   Label5.Caption = "Traida de instruccion"
   Label5.Visible = True
   Label5.ForeColor = QBColor(4)
   txtDirDat_P0.Visible = True
   imgArrow1.Picture = frmImagenes.imgLeftarrow1.Picture
   imgArrow1.Move 8040, 2040
   imgArrow1.Visible = True
Else
   Label5.Caption = "Lectura no válida"
   Label5.Visible = True
End If
P0_Eprom_QBcolor1
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 0.7 * tiem: Loop
txtDirDat_P0.Move 5640, 2660
imgArrow1.Move 5880, 2690
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 0.6 * tiem: Loop
txtDirDat_P0.Move 2760, 2180
imgArrow1.Move 2520, 2220
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 0.7 * tiem: Loop
txtDirDat_P0.Visible = False
imgArrow1.Visible = False
imgArrow1.Picture = frmImagenes.imgRightarrow1.Picture
Label4.Visible = False
Label5.Visible = False
frmDiagrama2.Refresh
txtDir_P2.Move 2760, 1190
txtDirDat_P0.Move 2760, 2180
If icolor1 = 2 Or icolor1 = 4 Or iclor1 = 6 Or icolor1 = 8 Then
   Form2.Temp_Cont_0
   Form2.Temp_Cont_1
End If
Next icolor1
label1.Visible = False
Label2.Visible = False
Label3.Visible = False
frmDiagrama2.Refresh
Form2.Atencion_Interrupciones
End Sub

Sub Byte2_1Ciclo_1()
EscrituraP1
LecturaP1
'Cada fase tendrá una duración de 2tiem segundo
tiem = Frecuencia_Oscilador
icolor = 16777215  'Line1.BorderColor
icolor2 = 16777215  'Shape1.BackColor
```

```
icolor3 = 8421504   'Shape4.BackColor
label1.Caption = saMatrizInstrucciones(ejec1)
Label2.Caption = "#Bytes = " & sInstruccionesOperandos(ejec1, 8)
Label3.Caption = "#C.Maq = " & sInstruccionesOperandos(ejec1, 9)
label1.Visible = True
Label2.Visible = True
Label3.Visible = True
For icolor1 = 1 To 2
txtDir_P2.Text = sInstruccionesOperandos(ejec1, 19 + 2 * (icolor1 - 1))
txtDirDat_P0.Text = sInstruccionesOperandos(ejec1, 20 + 2 * (icolor1 - 1))
'Segunda fase primer estado, primera fase segundo estado,
ALE_Latch_QBcolor1
PSEN_Eprom_QBcolor1
P2_Eprom_QBcolor1
P0_Eprom_color1
'Para Puerto P1 en WR
   If icolor1 = 1 And ejec1 > 0 And iEscrituraP1 Then
      If sInstruccionesOperandos(ejec1 - 1, 2) = "144" Or Mid(sInstruccionesOperandos(ejec1 - 1, 2), 1,
                                                2) = "P1" Then
         Puerto_P1_QBcolor1
      End If
   End If
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
P2_Eprom_color1
'Para Puerto P1 en WR
   If icolor1 = 1 And ejec1 > 0 And iEscrituraP1 Then
      If sInstruccionesOperandos(ejec1 - 1, 2) = "144" Then
         txtPuerto_P1.Text = sMemoriaRAMintH(144)
         txtPuerto_P1.Visible = True
         imgArrow_P1.Visible = True
      ElseIf Mid(sInstruccionesOperandos(ejec1 - 1, 2), 1, 2) = "P1" Then
         txtPuerto_P1.ForeColor = QBColor(12)
         txtPuerto_P1.Text = Mid(sInstruccionesOperandos(ejec1 - 1, 2), 3, 4)
         txtPuerto_P1.Visible = True
         imgArrow_P1.Visible = True
      End If
   End If
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
'Para Puerto P1 en WR
   If icolor1 = 1 And ejec1 > 0 And iEscrituraP1 Then
      If sInstruccionesOperandos(ejec1 - 1, 2) = "144" Or Mid(sInstruccionesOperandos(ejec1 - 1, 2), 1,
                                                2) = "P1" Then
         txtPuerto_P1 = ""
         txtPuerto_P1.Visible = False
         imgArrow_P1.Visible = False
         txtPuerto_P1.ForeColor = QBColor(0)
         Form2.Puerto_P1WR
         Puerto_P1_color1
      End If
   End If
P2_Eprom_QBcolor1
P0_Latch_QBcolor1
Label4.Caption = "Direccionamiento a ROM" & " " & "(" & icolor1 & " " & "Byte)"
Label4.Visible = True
Label4.ForeColor = QBColor(4)
txtDir_P2.Visible = True
txtDirDat_P0.Visible = True
imgArrow1.Visible = True
imgArrow2.Visible = True
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
'Envio del P2 y P0 como dirrecciones
txtDir_P2.Move 6120, 150
txtDirDat_P0.Move 5520, 1640
imgArrow1.Move 5760, 1680
imgArrow2.Move 5880, 160
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
'Segunda fase segundo estado
```

```
ALE_Latch_color1
PSEN_Eprom_QBcolor1
Latch_Eprom_QBcolor1
txtDir_P2.Move 9480, 150
txtDirDat_P0.Move 7320, 1640
imgArrow1.Move 7080, 1680
imgArrow2.Move 9240, 160
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
txtDir_P2.Move 9600, 630
txtDirDat_P0.Move 7800, 920
imgArrow1.Move 8040, 960
imgArrow2.Move 9360, 660
imgArrow2.Picture = frmImagenes.imgLeftarrow1.Picture
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
txtDir_P2.Visible = False
txtDirDat_P0.Visible = False
imgArrow1.Visible = False
imgArrow2.Visible = False
imgArrow2.Picture = frmImagenes.imgRightarrow1.Picture
imgArrow2.Move 2520, 1200
Label4.ForeColor = QBColor(0)
'Primera fase tercer estado, segunda fase tercer estado,
'primera fase cuarto estado
PSEN_Eprom_color1
P0_Latch_color1
Latch_Eprom_color1
frmDiagrama2.Refresh
'Para lectura de los interruptores del Puerto P1
If icolor1 = 2 Then
   Start = Timer: Do While Timer < Start + tiem * 2: Loop
   If sInstruccionesOperandos(ejec1, 3) = "144" Or iLecturaP1 = 1 Then
      'Para lectura de todos los bits
      Puerto_P1_QBcolor1
      frmDiagrama2.Refresh
      Start = Timer: Do While Timer < Start + tiem: Loop
      txtPuerto_P1.Text = sValor_InH_P1
      imgArrow_P1.Picture = frmImagenes.imgLeftarrow1.Picture
      txtPuerto_P1.Visible = True
      imgArrow_P1.Visible = True
      frmDiagrama2.Refresh
      Start = Timer: Do While Timer < Start + tiem: Loop
      txtPuerto_P1 = ""
      txtPuerto_P1.Visible = False
      imgArrow_P1.Visible = False
      imgArrow_P1.Picture = frmImagenes.imgRightarrow1.Picture
      Puerto_P1_color1
      Form2.Puerto_P1RD
   ElseIf Mid(sInstruccionesOperandos(ejec1, 3), 1, 2) = "P1" _
         Or Mid(sInstruccionesOperandos(ejec1, 3), 1, 3) = "/P1" Or iLecturaP1 = 2 Then
      'Para de uno de los 4 pines de entrada en  P1
      Puerto_P1_QBcolor1
      frmDiagrama2.Refresh
      Start = Timer: Do While Timer < Start + tiem: Loop
      txtPuerto_P1.ForeColor = QBColor(12)
      If iLecturaP1 = 2 Then
         txtPuerto_P1.Text = Mid(sInstruccionesOperandos(ejec1, 2), 3, 4)
      Else
         txtPuerto_P1.Text = Mid(sInstruccionesOperandos(ejec1, 3), 3, 4)
      End If
      imgArrow_P1.Picture = frmImagenes.imgLeftarrow1.Picture
      txtPuerto_P1.Visible = True
      imgArrow_P1.Visible = True
      frmDiagrama2.Refresh
      Start = Timer: Do While Timer < Start + tiem: Loop
      txtPuerto_P1.ForeColor = QBColor(0)
      txtPuerto_P1 = ""
      txtPuerto_P1.Visible = False
      imgArrow_P1.Visible = False
      imgArrow_P1.Picture = frmImagenes.imgRightarrow1.Picture
```

```
        Puerto_P1_color1
        Form2.Puerto_P1RD
    Else
        Start = Timer: Do While Timer < Start + tiem * 2: Loop
    End If
Else
    Start = Timer: Do While Timer < Start + tiem * 4: Loop
End If
txtDirDat_P0.Move 7800, 2010
imgArrow1.Picture = frmImagenes.imgLeftarrow1.Picture
imgArrow1.Move 8040, 2040
Label5.Caption = "Traida de instruccion" & " " & "(" & icolor1 & " " & "Byte)"
Label5.Visible = True
Label5.ForeColor = QBColor(4)
txtDirDat_P0.Text = sInstruccionesOperandos(ejec1, 16 + icolor1 - 1)
txtDirDat_P0.Visible = True
imgArrow1.Visible = True
P0_Eprom_QBcolor1
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 0.7 * tiem: Loop
txtDirDat_P0.Move 5640, 2660
imgArrow1.Move 5880, 2690
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 0.6 * tiem: Loop
txtDirDat_P0.Move 2760, 2180
imgArrow1.Move 2520, 2220
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 0.7 * tiem: Loop
txtDirDat_P0.Visible = False
imgArrow1.Visible = False
imgArrow1.Picture = frmImagenes.imgRightarrow1.Picture
Label4.Visible = False
Label5.Visible = False
frmDiagrama2.Refresh
txtDir_P2.Move 2760, 1190
txtDirDat_P0.Move 2760, 2180
Next icolor1
'Finaliza el único ciclo de maquina
Form2.Temp_Cont_0
Form2.Temp_Cont_1
label1.Visible = False
Label2.Visible = False
Label3.Visible = False
frmDiagrama2.Refresh
Form2.Atencion_Interrupciones
End Sub

Sub Byte2_2Ciclo_1()
EscrituraP1
LecturaP1
'Cada fase tendrá una duración de 2tiem segundo
tiem = Frecuencia_Oscilador
icolor = 16777215   'Line1.BorderColor
icolor2 = 16777215  'Shape1.BackColor
icolor3 = 8421504   'Shape4.BackColor
label1.Caption = saMatrizInstrucciones(ejec1)
Label2.Caption = "#Bytes = " & sInstruccionesOperandos(ejec1, 8)
Label3.Caption = "#C.Maq = " & sInstruccionesOperandos(ejec1, 9)
label1.Visible = True
Label2.Visible = True
Label3.Visible = True
For icolor1 = 1 To 4
If icolor1 = 1 Or icolor1 = 2 Then
    txtDir_P2.Text = sInstruccionesOperandos(ejec1, 19 + 2 * (icolor1 - 1))
    txtDirDat_P0.Text = sInstruccionesOperandos(ejec1, 20 + 2 * (icolor1 - 1))
ElseIf icolor1 = 3 Or icolor1 = 4 Then
    txtDirDat_P0.Text = sInstruccionesOperandos(ejec1, 20 + 2)
End If
'Segunda fase primer estado, primera fase segundo estado,
ALE_Latch_QBcolor1
PSEN_Eprom_QBcolor1
```

```
P2_Eprom_QBcolor1
P0_Eprom_color1
'Para Puerto P1 en WR
   If icolor1 = 1 And ejec1 > 0 And iEscrituraP1 Then
      If sInstruccionesOperandos(ejec1 - 1, 2) = "144" Or Mid(sInstruccionesOperandos(ejec1 - 1, 2), 1,
                                                    2) = "P1" Then
         Puerto_P1_QBcolor1
      End If
   End If
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
P2_Eprom_color1
'Para Puerto P1 en WR
   If icolor1 = 1 And ejec1 > 0 And iEscrituraP1 Then
      If sInstruccionesOperandos(ejec1 - 1, 2) = "144" Then
         txtPuerto_P1.Text = sMemoriaRAMintH(144)
         txtPuerto_P1.Visible = True
         imgArrow_P1.Visible = True
      ElseIf Mid(sInstruccionesOperandos(ejec1 - 1, 2), 1, 2) = "P1" Then
         txtPuerto_P1.ForeColor = QBColor(12)
         txtPuerto_P1.Text = Mid(sInstruccionesOperandos(ejec1 - 1, 2), 3, 4)
         txtPuerto_P1.Visible = True
         imgArrow_P1.Visible = True
      End If
   End If
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
'Para Puerto P1 en WR
   If icolor1 = 1 And ejec1 > 0 And iEscrituraP1 Then
      If sInstruccionesOperandos(ejec1 - 1, 2) = "144" Or Mid(sInstruccionesOperandos(ejec1 - 1, 2), 1,
                                                    2) = "P1" Then
         txtPuerto_P1 = ""
         txtPuerto_P1.Visible = False
         imgArrow_P1.Visible = False
         txtPuerto_P1.ForeColor = QBColor(0)
         Form2.Puerto_P1WR
         Puerto_P1_color1
      End If
   End If
P2_Eprom_QBcolor1
P0_Latch_QBcolor1
If icolor1 = 1 Or icolor1 = 2 Then
   Label4.Caption = "Direccionamiento a ROM" & " " & "(" & icolor1 & " " & "Byte)"
Else
   Label4.Caption = "Direccionamiento a ROM"
End If
Label4.Visible = True
Label4.ForeColor = QBColor(4)
txtDir_P2.Visible = True
txtDirDat_P0.Visible = True
imgArrow1.Visible = True
imgArrow2.Visible = True
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
'Envio del P2 y P0 como dirrecciones
txtDir_P2.Move 6120, 150
txtDirDat_P0.Move 5520, 1640
imgArrow1.Move 5760, 1680
imgArrow2.Move 5880, 160
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
'Segunda fase segundo estado
ALE_Latch_color1
PSEN_Eprom_QBcolor1
Latch_Eprom_QBcolor1
txtDir_P2.Move 9480, 150
txtDirDat_P0.Move 7320, 1640
imgArrow1.Move 7080, 1680
imgArrow2.Move 9240, 160
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
```

```
txtDir_P2.Move 9600, 630
txtDirDat_P0.Move 7800, 920
imgArrow1.Move 8040, 960
imgArrow2.Picture = frmImagenes.imgLeftarrow1.Picture
imgArrow2.Move 9360, 660
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
txtDir_P2.Visible = False
txtDirDat_P0.Visible = False
imgArrow1.Visible = False
imgArrow2.Visible = False
imgArrow2.Picture = frmImagenes.imgRightarrow1.Picture
imgArrow2.Move 2520, 1200
Label4.ForeColor = QBColor(0)
frmDiagrama2.Refresh
'Primera fase tercer estado, segunda fase tercer estado,
'primera fase cuarto estado
PSEN_Eprom_color1
P0_Latch_color1
Latch_Eprom_color1
frmDiagrama2.Refresh
'Para lectura de los interruptores del Puerto P1
If icolor1 = 4 Then
   Start = Timer: Do While Timer < Start + tiem * 2: Loop
   If sInstruccionesOperandos(ejec1, 3) = "144" Or iLecturaP1 = 1 Then
      'Para lectura de todos los bits
      Puerto_P1_QBcolor1
      frmDiagrama2.Refresh
      Start = Timer: Do While Timer < Start + tiem: Loop
      txtPuerto_P1.Text = sValor_InH_P1
      imgArrow_P1.Picture = frmImagenes.imgLeftarrow1.Picture
      txtPuerto_P1.Visible = True
      imgArrow_P1.Visible = True
      frmDiagrama2.Refresh
      Start = Timer: Do While Timer < Start + tiem: Loop
      txtPuerto_P1 = ""
      txtPuerto_P1.Visible = False
      imgArrow_P1.Visible = False
      imgArrow_P1.Picture = frmImagenes.imgRightarrow1.Picture
      Puerto_P1_color1
      Form2.Puerto_P1RD
   ElseIf Mid(sInstruccionesOperandos(ejec1, 3), 1, 2) = "P1" _
         Or Mid(sInstruccionesOperandos(ejec1, 3), 1, 3) = "/P1" Or iLecturaP1 = 2 Then
         'And sInstruccionesOperandos(ejec1, 1) = "MOV"
      'Para de uno de los 4 pines de entrada en  P1
      Puerto_P1_QBcolor1
      frmDiagrama2.Refresh
      Start = Timer: Do While Timer < Start + tiem: Loop
      txtPuerto_P1.ForeColor = QBColor(12)
      If iLecturaP1 = 2 Then
         txtPuerto_P1.Text = Mid(sInstruccionesOperandos(ejec1, 2), 3, 4)
      Else
         txtPuerto_P1.Text = Mid(sInstruccionesOperandos(ejec1, 3), 3, 4)
      End If
      imgArrow_P1.Picture = frmImagenes.imgLeftarrow1.Picture
      txtPuerto_P1.Visible = True
      imgArrow_P1.Visible = True
      frmDiagrama2.Refresh
      Start = Timer: Do While Timer < Start + tiem: Loop
      txtPuerto_P1.ForeColor = QBColor(0)
      txtPuerto_P1 = ""
      txtPuerto_P1.Visible = False
      imgArrow_P1.Visible = False
      imgArrow_P1.Picture = frmImagenes.imgRightarrow1.Picture
      Puerto_P1_color1
      Form2.Puerto_P1RD
   Else
      Start = Timer: Do While Timer < Start + tiem * 2: Loop
   End If
Else
   Start = Timer: Do While Timer < Start + tiem * 4: Loop
```

```
End If
txtDirDat_P0.Move 7800, 2010
If icolor1 = 1 Or icolor1 = 2 Then
   Label5.Caption = "Traida de instruccion" & " " & "(" & icolor1 & " " & "Byte)"
   txtDirDat_P0.Text = sInstruccionesOperandos(ejec1, 16 + icolor1 - 1)
   imgArrow1.Picture = frmImagenes.imgLeftarrow1.Picture
   imgArrow1.Move 8040, 2040
   txtDirDat_P0.Visible = True
   imgArrow1.Visible = True
Else
   Label5.Caption = "Lectura no válida"
End If
Label5.Visible = True
Label5.ForeColor = QBColor(4)
P0_Eprom_QBcolor1
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 0.7 * tiem: Loop
txtDirDat_P0.Move 5640, 2660
imgArrow1.Move 5880, 2690
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 0.6 * tiem: Loop
txtDirDat_P0.Move 2760, 2180
imgArrow1.Move 2520, 2220
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 0.7 * tiem: Loop
txtDirDat_P0.Visible = False
imgArrow1.Visible = False
imgArrow1.Picture = frmImagenes.imgRightarrow1.Picture
Label4.Visible = False
Label5.Visible = False
frmDiagrama2.Refresh
txtDir_P2.Move 2760, 1190
txtDirDat_P0.Move 2760, 2180
If icolor1 = 2 Or icolor1 = 4 Then
   Form2.Temp_Cont_0
   Form2.Temp_Cont_1
End If
Next icolor1
label1.Visible = False
Label2.Visible = False
Label3.Visible = False
frmDiagrama2.Refresh
Form2.Atencion_Interrupciones
End Sub

Sub Byte3_2Ciclo_1()
EscrituraP1
LecturaP1
'Cada fase tendrá una duración de 2tiem segundo
tiem = Frecuencia_Oscilador
icolor = 16777215   'Line1.BorderColor
icolor2 = 16777215  'Shape1.BackColor
icolor3 = 8421504   'Shape4.BackColor
label1.Caption = saMatrizInstrucciones(ejec1)
Label2.Caption = "#Bytes = " & sInstruccionesOperandos(ejec1, 8)
Label3.Caption = "#C.Maq = " & sInstruccionesOperandos(ejec1, 9)
label1.Visible = True
Label2.Visible = True
Label3.Visible = True
For icolor1 = 1 To 4
If icolor1 = 1 Or icolor1 = 2 Or icolor1 = 3 Then
   txtDir_P2.Text = sInstruccionesOperandos(ejec1, 19 + 2 * (icolor1 - 1))
   txtDirDat_P0.Text = sInstruccionesOperandos(ejec1, 20 + 2 * (icolor1 - 1))
ElseIf icolor1 = 4 Then
   txtDirDat_P0.Text = sInstruccionesOperandos(ejec1, 20 + 4)
End If
'Segunda fase primer estado, primera fase segundo estado,
ALE_Latch_QBcolor1
PSEN_Eprom_QBcolor1
P2_Eprom_QBcolor1
P0_Eprom_color1
```

```
'Para Puerto P1 en WR
   If icolor1 = 1 And ejec1 > 0 And iEscrituraP1 Then
      If sInstruccionesOperandos(ejec1 - 1, 2) = "144" Or Mid(sInstruccionesOperandos(ejec1 - 1, 2), 1,
                                                    2) = "P1" Then
         Puerto_P1_QBcolor1
      End If
   End If
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
P2_Eprom_color1
'Para Puerto P1 en WR
   If icolor1 = 1 And ejec1 > 0 And iEscrituraP1 Then
      If sInstruccionesOperandos(ejec1 - 1, 2) = "144" Then
         txtPuerto_P1.Text = sMemoriaRAMintH(144)
         txtPuerto_P1.Visible = True
         imgArrow_P1.Visible = True
      ElseIf Mid(sInstruccionesOperandos(ejec1 - 1, 2), 1, 2) = "P1" Then
         txtPuerto_P1.ForeColor = QBColor(12)
         txtPuerto_P1.Text = Mid(sInstruccionesOperandos(ejec1 - 1, 2), 3, 4)
         txtPuerto_P1.Visible = True
         imgArrow_P1.Visible = True
      End If
   End If
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
'Para Puerto P1 en WR
   If icolor1 = 1 And ejec1 > 0 And iEscrituraP1 Then
      If sInstruccionesOperandos(ejec1 - 1, 2) = "144" Or Mid(sInstruccionesOperandos(ejec1 - 1, 2), 1,
                                                    2) = "P1" Then
         txtPuerto_P1 = ""
         txtPuerto_P1.Visible = False
         imgArrow_P1.Visible = False
         txtPuerto_P1.ForeColor = QBColor(0)
         Form2.Puerto_P1WR
         Puerto_P1_color1
      End If
   End If
P2_Eprom_QBcolor1
P0_Latch_QBcolor1
If icolor1 = 1 Or icolor1 = 2 Or icolor1 = 3 Then
   Label4.Caption = "Direccionamiento a ROM" & " " & "(" & icolor1 & " " & "Byte)"
Else
   Label4.Caption = "Direccionamiento a ROM"
End If
Label4.Visible = True
Label4.ForeColor = QBColor(4)
txtDir_P2.Visible = True
txtDirDat_P0.Visible = True
imgArrow1.Visible = True
imgArrow2.Visible = True
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
'Envio del P2 y P0 como dirrecciones
txtDir_P2.Move 6120, 150
txtDirDat_P0.Move 5520, 1640
imgArrow1.Move 5760, 1680
imgArrow2.Move 5880, 160
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
'Segunda fase segundo estado
ALE_Latch_color1
PSEN_Eprom_QBcolor1
Latch_Eprom_QBcolor1
txtDir_P2.Move 9480, 150
txtDirDat_P0.Move 7320, 1640
imgArrow1.Move 7080, 1680
imgArrow2.Move 9240, 160
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
txtDir_P2.Move 9600, 630
txtDirDat_P0.Move 7800, 920
```

```
imgArrow1.Move 8040, 960
imgArrow2.Picture = frmImagenes.imgLeftarrow1.Picture
imgArrow2.Move 9360, 660
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + tiem: Loop
txtDir_P2.Visible = False
txtDirDat_P0.Visible = False
imgArrow1.Visible = False
imgArrow2.Visible = False
imgArrow2.Picture = frmImagenes.imgRightarrow1.Picture
imgArrow2.Move 2520, 1200
Label4.ForeColor = QBColor(0)
frmDiagrama2.Refresh
frmDiagrama2.Refresh
'Primera fase tercer estado, segunda fase tercer estado,
'primera fase cuarto estado
ALE_Latch_color1
PSEN_Eprom_color1
P0_Latch_color1
Latch_Eprom_color1
frmDiagrama2.Refresh
If icolor1 = 4 Then
   Start = Timer: Do While Timer < Start + tiem * 2: Loop
   If sInstruccionesOperandos(ejec1, 3) = "144" Or iLecturaP1 = 1 Then
      'Para lectura de todos los bits
      Puerto_P1_QBcolor1
      frmDiagrama2.Refresh
      Start = Timer: Do While Timer < Start + tiem: Loop
      txtPuerto_P1.Text = sValor_InH_P1
      imgArrow_P1.Picture = frmImagenes.imgLeftarrow1.Picture
      txtPuerto_P1.Visible = True
      imgArrow_P1.Visible = True
      frmDiagrama2.Refresh
      Start = Timer: Do While Timer < Start + tiem: Loop
      txtPuerto_P1 = ""
      txtPuerto_P1.Visible = False
      imgArrow_P1.Visible = False
      imgArrow_P1.Picture = frmImagenes.imgRightarrow1.Picture
      Puerto_P1_color1
      Form2.Puerto_P1RD
   ElseIf Mid(sInstruccionesOperandos(ejec1, 3), 1, 2) = "P1" _
         Or Mid(sInstruccionesOperandos(ejec1, 3), 1, 3) = "/P1" Or iLecturaP1 = 2 Then
         'And sInstruccionesOperandos(ejec1, 1) = "MOV"
      'Para de uno de los 4 pines de entrada en  P1
      Puerto_P1_QBcolor1
      frmDiagrama2.Refresh
      Start = Timer: Do While Timer < Start + tiem: Loop
      txtPuerto_P1.ForeColor = QBColor(12)
      If iLecturaP1 = 2 Then
         txtPuerto_P1.Text = Mid(sInstruccionesOperandos(ejec1, 2), 3, 4)
      Else
         txtPuerto_P1.Text = Mid(sInstruccionesOperandos(ejec1, 3), 3, 4)
      End If
      imgArrow_P1.Picture = frmImagenes.imgLeftarrow1.Picture
      txtPuerto_P1.Visible = True
      imgArrow_P1.Visible = True
      frmDiagrama2.Refresh
      Start = Timer: Do While Timer < Start + tiem: Loop
      txtPuerto_P1.ForeColor = QBColor(0)
      txtPuerto_P1 = ""
      txtPuerto_P1.Visible = False
      imgArrow_P1.Visible = False
      imgArrow_P1.Picture = frmImagenes.imgRightarrow1.Picture
      Puerto_P1_color1
      Form2.Puerto_P1RD
   Else
      Start = Timer: Do While Timer < Start + tiem * 2: Loop
   End If
Else
   Start = Timer: Do While Timer < Start + tiem * 4: Loop
End If
```

```
txtDirDat_P0.Move 7800, 2010
If icolor1 = 1 Or icolor1 = 2 Or icolor1 = 3 Then
   Label5.Caption = "Traida de instruccion" & " " & "(" & icolor1 & " " & "Byte)"
   txtDirDat_P0.Text = sInstruccionesOperandos(ejec1, 16 + icolor1 - 1)
   imgArrow1.Picture = frmImagenes.imgLeftarrow1.Picture
   imgArrow1.Move 8040, 2040
   txtDirDat_P0.Visible = True
   imgArrow1.Visible = True
Else
   Label5.Caption = "Lectura no válida"
End If
Label5.Visible = True
Label5.ForeColor = QBColor(4)
P0_Eprom_QBcolor1
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 0.7 * tiem: Loop
txtDirDat_P0.Move 5640, 2660
imgArrow1.Move 5880, 2690
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 0.6 * tiem: Loop
txtDirDat_P0.Move 2760, 2180
imgArrow1.Move 2520, 2220
frmDiagrama2.Refresh
Start = Timer: Do While Timer < Start + 0.7 * tiem: Loop
txtDirDat_P0.Visible = False
imgArrow1.Visible = False
imgArrow1.Picture = frmImagenes.imgRightarrow1.Picture
Label4.Visible = False
Label5.Visible = False
frmDiagrama2.Refresh
txtDir_P2.Move 2760, 1190
txtDirDat_P0.Move 2760, 2180
If icolor1 = 2 Or icolor1 = 4 Then
   Form2.Temp_Cont_0
   Form2.Temp_Cont_1
End If
Next icolor1
label1.Visible = False
Label2.Visible = False
Label3.Visible = False
frmDiagrama2.Refresh
Form2.Atencion_Interrupciones
End Sub

Sub EscrituraP1()
If ejec1 > 0 Then
   If sInstruccionesOperandos(ejec1 - 1, 1) <> "PUSH" _
      And sInstruccionesOperandos(ejec1 - 1, 1) <> "JB" _
      And sInstruccionesOperandos(ejec1 - 1, 1) <> "JNB" Then
      iEscrituraP1 = True
   Else
      iEscrituraP1 = False
   End If
Else
   iEscrituraP1 = False
End If
End Sub

Sub Latch_Eprom_color1()
Shape4.BackColor = icolor3
Shape5.BackColor = icolor3
Shape4.BorderColor = icolor3
Shape5.BorderColor = icolor3
Line32.BorderColor = icolor3
Line33.BorderColor = icolor3
Line34.BorderColor = icolor3
End Sub

Sub Latch_Eprom_QBcolor1()
Shape4.BackColor = QBColor(3)
Shape5.BackColor = QBColor(3)
```

```
Shape4.BorderColor = QBColor(3)
Shape5.BorderColor = QBColor(3)
Line32.BorderColor = QBColor(3)
Line33.BorderColor = QBColor(3)
Line34.BorderColor = QBColor(3)
End Sub

Sub Latch_Ram_color1()
Shape4.BackColor = icolor3
Shape8.BackColor = icolor3
Shape4.BorderColor = icolor3
Shape8.BorderColor = icolor3
Line32.BorderColor = icolor3
Line42.BorderColor = icolor3
Line43.BorderColor = icolor3
End Sub

Sub Latch_Ram_QBcolor1()
Shape4.BackColor = QBColor(3)
Shape8.BackColor = QBColor(3)
Shape4.BorderColor = QBColor(3)
Shape8.BorderColor = QBColor(3)
Line32.BorderColor = QBColor(3)
Line42.BorderColor = QBColor(3)
Line43.BorderColor = QBColor(3)
End Sub

Sub LecturaP1()
If sInstruccionesOperandos(ejec1, 1) = "PUSH" _
  And sInstruccionesOperandos(ejec1, 2) = "P1" Then
   iLecturaP1 = 1 'Si es byte
ElseIf (sInstruccionesOperandos(ejec1, 1) = "JB" _
     Or sInstruccionesOperandos(ejec1, 1) = "JNB") _
     And (Mid(sInstruccionesOperandos(ejec1, 2), 1, 2) = "P1" _
     Or Mid(sInstruccionesOperandos(ejec1, 2), 1, 3) = "/P1") Then
      iLecturaP1 = 2 'Si es bit
Else
   iLecturaP1 = 5  'Un valor cualquiera diferente de 1 y 2
End If
End Sub

Sub P0_Buffer_color1()
Shape2.BackColor = icolor2
Shape15.BackColor = icolor2
Shape2.BorderColor = icolor2
Shape15.BorderColor = icolor2
Line29.BorderColor = icolor2
Line35.BorderColor = icolor2
Line36.BorderColor = icolor2
Line40.BorderColor = icolor2
Line56.BorderColor = icolor2
End Sub

Sub P0_Buffer_QBcolor1()
Shape2.BackColor = QBColor(6)
Shape15.BackColor = QBColor(6)
Shape2.BorderColor = QBColor(6)
Shape15.BorderColor = QBColor(6)
Line29.BorderColor = QBColor(6)
Line35.BorderColor = QBColor(6)
Line36.BorderColor = QBColor(6)
Line40.BorderColor = QBColor(6)
Line56.BorderColor = QBColor(6)
End Sub

Sub P0_Eprom_color1()
Shape7.BackColor = icolor2
Shape2.BackColor = icolor2
Shape7.BorderColor = icolor2
Shape2.BorderColor = icolor2
Line39.BorderColor = icolor2
```

```
Line38.BorderColor = icolor2
Line37.BorderColor = icolor2
Line36.BorderColor = icolor2
Line35.BorderColor = icolor2
Line29.BorderColor = icolor2
End Sub

Sub P0_Eprom_QBcolor1()
Shape7.BackColor = QBColor(6)
Shape2.BackColor = QBColor(6)
Shape7.BorderColor = QBColor(6)
Shape2.BorderColor = QBColor(6)
Line39.BorderColor = QBColor(6)
Line38.BorderColor = QBColor(6)
Line37.BorderColor = QBColor(6)
Line36.BorderColor = QBColor(6)
Line35.BorderColor = QBColor(6)
Line29.BorderColor = QBColor(6)
End Sub

Sub P0_Latch_color1()
Shape2.BackColor = icolor2
Shape3.BackColor = icolor2
Shape2.BorderColor = icolor2
Shape3.BorderColor = icolor2
Line29.BorderColor = icolor2
Line30.BorderColor = icolor2
Line31.BorderColor = icolor2
End Sub

Sub P0_Latch_QBcolor1()
Shape2.BackColor = QBColor(3)
Shape3.BackColor = QBColor(3)
Shape2.BorderColor = QBColor(3)
Shape3.BorderColor = QBColor(3)
Line29.BorderColor = QBColor(3)
Line30.BorderColor = QBColor(3)
Line31.BorderColor = QBColor(3)
End Sub

Sub P0_Latch377_color1()
Shape2.BackColor = icolor2
Shape14.BackColor = icolor2
Shape2.BorderColor = icolor2
Shape14.BorderColor = icolor2
Line29.BorderColor = icolor2
Line35.BorderColor = icolor2.BorderColor = icolor2
Line40.BorderColor = icolor2
Line50.BorderColor = icolor2
Line54.BorderColor = icolor2
Line55.BorderColor = icolor2
End Sub

Sub P0_Latch377_QBcolor1()
Shape2.BackColor = QBColor(6)
Shape14.BackColor = QBColor(6)
Shape2.BorderColor = QBColor(6)
Shape14.BorderColor = QBColor(6)
Line29.BorderColor = QBColor(6)
Line35.BorderColor = QBColor(6)
Line36.BorderColor = QBColor(6)
Line40.BorderColor = QBColor(6)
Line50.BorderColor = QBColor(6)
Line54.BorderColor = QBColor(6)
Line55.BorderColor = QBColor(6)
End Sub

Sub P0_NoAdd_color1()
Shape2.BackColor = icolor2
Shape2.BorderColor = icolor2
Line29.BorderColor = icolor2
```

```
Line35.BorderColor = icolor2
Line36.BorderColor = icolor2
Line40.BorderColor = icolor2
End Sub

Sub P0_NoAdd_QBcolor1()
Shape2.BackColor = QBColor(6)
Shape2.BorderColor = QBColor(6)
Line29.BorderColor = QBColor(6)
Line35.BorderColor = QBColor(6)
Line36.BorderColor = QBColor(6)
Line40.BorderColor = QBColor(6)
End Sub

Sub P0_NoMOVC_color1()
Shape2.BackColor = icolor2
Shape2.BorderColor = icolor2
Line37.BorderColor = icolor2
Line36.BorderColor = icolor2
Line35.BorderColor = icolor2
Line29.BorderColor = icolor2
End Sub

Sub P0_NoMOVC_QBcolor1()
Shape2.BackColor = QBColor(6)
Shape2.BorderColor = QBColor(6)
Line37.BorderColor = QBColor(6)
Line36.BorderColor = QBColor(6)
Line35.BorderColor = QBColor(6)
Line29.BorderColor = QBColor(6)
End Sub

Sub P0_Ram_color1()
Shape2.BackColor = icolor2
Shape10.BackColor = icolor2
Shape2.BorderColor = icolor2
Shape10.BorderColor = icolor2
Line29.BorderColor = icolor2
Line35.BorderColor = icolor2
Line36.BorderColor = icolor2
Line40.BorderColor = icolor2
Line41.BorderColor = icolor2
Line50.BorderColor = icolor2
End Sub

Sub P0_Ram_QBcolor1()
Shape2.BackColor = QBColor(6)
Shape10.BackColor = QBColor(6)
Shape2.BorderColor = QBColor(6)
Shape10.BorderColor = QBColor(6)
Line29.BorderColor = QBColor(6)
Line35.BorderColor = QBColor(6)
Line36.BorderColor = QBColor(6)
Line40.BorderColor = QBColor(6)
Line41.BorderColor = QBColor(6)
Line50.BorderColor = QBColor(6)
End Sub

Sub P2_Eprom_color1()
Shape1.BackColor = icolor2
Shape6.BackColor = icolor2
Shape1.BorderColor = icolor2
Shape6.BorderColor = icolor2
Line22.BorderColor = icolor2
Line23.BorderColor = icolor2
Line24.BorderColor = icolor2
Line25.BorderColor = icolor2
Line26.BorderColor = icolor2
'Para el Decoder
Shape12.BackColor = icolor2
Shape12.BorderColor = icolor2
```

```
Shape13.BackColor = icolor2
Shape13.BorderColor = icolor2
Line45.BorderColor = icolor2
Line46.BorderColor = icolor2
Line47.BorderColor = icolor2
lblDecoder.Visible = False
Line53.Visible = False
lblDec_Salida.Visible = False
Label14.Visible = False
Line60.Visible = False
End Sub

Sub P2_Eprom_QBcolor1()
Shape1.BackColor = QBColor(3)
Shape6.BackColor = QBColor(3)
Shape1.BorderColor = QBColor(3)
Shape6.BorderColor = QBColor(3)
Line22.BorderColor = QBColor(3)
Line23.BorderColor = QBColor(3)
Line24.BorderColor = QBColor(3)
Line25.BorderColor = QBColor(3)
Line26.BorderColor = QBColor(3)
'Para el Decoder
Shape12.BackColor = QBColor(3)
Shape12.BorderColor = QBColor(3)
Shape13.BackColor = QBColor(3)
Shape13.BorderColor = QBColor(3)
Line45.BorderColor = QBColor(3)
Line46.BorderColor = QBColor(3)
Line47.BorderColor = QBColor(3)
lblDecoder.Caption = "0"
lblDecoder.Visible = True
Line53.Y1 = 1425
Line53.Y2 = 1425
lblDec_Salida.Caption = "EPROM"
lblDec_Salida.Top = 1320
lblDec_Salida.Left = 4440
Line53.Visible = True
lblDec_Salida.Visible = True
Label14.Visible = True
Line60.Visible = True
End Sub

Sub P2_Ram_color1()
Shape1.BackColor = icolor2
Shape9.BackColor = icolor2
Shape1.BorderColor = icolor2
Shape9.BorderColor = icolor2
Line22.BorderColor = icolor2
Line23.BorderColor = icolor2
Line24.BorderColor = icolor2
Line25.BorderColor = icolor2
Line27.BorderColor = icolor2
Line28.BorderColor = icolor2
'Para el Decoder
Shape12.BackColor = icolor2
Shape12.BorderColor = icolor2
Shape13.BackColor = icolor2
Shape13.BorderColor = icolor2
Line45.BorderColor = icolor2
Line46.BorderColor = icolor2
Line47.BorderColor = icolor2
lblDecoder.Visible = False
Line53.Visible = False
lblDec_Salida.Visible = False
Label13.Visible = False
Line59.Visible = False
Label12.Visible = False
Line58.Visible = False
Label11.Visible = False
Line57.Visible = False
```

```
End Sub

Sub P2_Ram_QBcolor1()
Shape1.BackColor = QBColor(3)
Shape9.BackColor = QBColor(3)
Shape1.BorderColor = QBColor(3)
Shape9.BorderColor = QBColor(3)
Line22.BorderColor = QBColor(3)
Line23.BorderColor = QBColor(3)
Line24.BorderColor = QBColor(3)
Line25.BorderColor = QBColor(3)
Line27.BorderColor = QBColor(3)
Line28.BorderColor = QBColor(3)
'Para el Decoder
Shape12.BackColor = QBColor(3)
Shape12.BorderColor = QBColor(3)
Shape13.BackColor = QBColor(3)
Shape13.BorderColor = QBColor(3)
Line45.BorderColor = QBColor(3)
Line46.BorderColor = QBColor(3)
Line47.BorderColor = QBColor(3)
Select Case iRamBufferLatch
  Case 1
    lblDecoder.Caption = iRam
    lblDecoder.Visible = True
    iRamBufferLatch2 = iRam
    RamBufferLatch
    lblDec_Salida.Caption = "RAM"
    lblDec_Salida.Left = 4440
    Label13.Visible = True
    Line59.Visible = True
  Case 2
    lblDecoder.Caption = iBuffer
    lblDecoder.Visible = True
    iRamBufferLatch2 = iBuffer
    RamBufferLatch
    lblDec_Salida.Caption = "Buffer"
    lblDec_Salida.Left = 4440
    Label12.Visible = True
    Line58.Visible = True
  Case 3
    lblDecoder.Caption = iLatch
    lblDecoder.Visible = True
    iRamBufferLatch2 = iLatch
    RamBufferLatch
    lblDec_Salida.Caption = "Latch"
    lblDec_Salida.Left = 4440
    Label11.Visible = True
    Line57.Visible = True
  Case 4 To 5   'Si no existe ningun elemento "RAM" validado
    lblDecoder.Caption = iRamBufferLatch1
    lblDecoder.Visible = True
    iRamBufferLatch2 = iRamBufferLatch1
    RamBufferLatch
    lblDec_Salida.Caption = "Ninguno"
    lblDec_Salida.Left = 4440
End Select
Line53.Visible = True
lblDec_Salida.Visible = True
End Sub

Sub PSEN_Eprom_color1()
Line4.BorderColor = icolor
Line5.BorderColor = icolor
Line6.BorderColor = icolor
Line7.BorderColor = icolor
End Sub

Sub PSEN_Eprom_QBcolor1()
Line4.BorderColor = QBColor(12)
Line5.BorderColor = QBColor(12)
```

```
Line6.BorderColor = QBColor(12)
Line7.BorderColor = QBColor(12)
End Sub

Sub Puerto_P1_color1()
Line44.BorderColor = icolor
Shape11.BorderColor = icolor
Shape11.BackColor = icolor
End Sub

Sub Puerto_P1_QBcolor1()
Line44.BorderColor = QBColor(6)
Shape11.BorderColor = QBColor(6)
Shape11.BackColor = QBColor(6)
End Sub

Sub RamBufferLatch()
Select Case iRamBufferLatch2
   Case 0
      Line53.Y1 = 1425
      Line53.Y2 = 1425
      lblDec_Salida.Top = 1320
   Case 1
      Line53.Y1 = 1515
      Line53.Y2 = 1515
      lblDec_Salida.Top = 1425
   Case 2
      Line53.Y1 = 1605
      Line53.Y2 = 1605
      lblDec_Salida.Top = 1515
   Case 3
      Line53.Y1 = 1710
      Line53.Y2 = 1710
      lblDec_Salida.Top = 1620
   Case 4
      Line53.Y1 = 1800
      Line53.Y2 = 1800
      lblDec_Salida.Top = 1710
   Case 5
      Line53.Y1 = 1905
      Line53.Y2 = 1905
      lblDec_Salida.Top = 1815
   Case 6
      Line53.Y1 = 1995
      Line53.Y2 = 1995
      lblDec_Salida.Top = 1905
   Case 7
      Line53.Y1 = 2085
      Line53.Y2 = 2085
      lblDec_Salida.Top = 1995
End Select
End Sub

Sub RD_Buffer_color1()
Line15.BorderColor = QBColor(12)
Line16.BorderColor = QBColor(12)
Line17.BorderColor = QBColor(12)
Line18.BorderColor = QBColor(12)
Line19.BorderColor = QBColor(12)
Line52.BorderColor = QBColor(12)
'A la entrada del Buffer
Shape16.BackColor = icolor2
Shape16.BorderColor = icolor2
Line61.BorderColor = icolor2
End Sub

Sub RD_Buffer_QBcolor1()
Line15.BorderColor = icolor
Line16.BorderColor = icolor
Line17.BorderColor = icolor
Line18.BorderColor = icolor
```

```
Line19.BorderColor = Icolor
Line52.BorderColor = icolor
'A la entrada del Buffer
Shape16.BackColor = QBColor(6)
Shape16.BorderColor = QBColor(6)
Line61.BorderColor = QBColor(6)
End Sub

Sub RD_NoAdd_color1()
Line15.BorderColor = QBColor(12)
Line16.BorderColor = QBColor(12)
Line17.BorderColor = QBColor(12)
Line18.BorderColor = QBColor(12)
Line19.BorderColor = QBColor(12)
Line48.BorderColor = QBColor(12)
End Sub

Sub RD_NoAdd_QBcolor1()
Line15.BorderColor = icolor
Line16.BorderColor = icolor
Line17.BorderColor = icolor
Line18.BorderColor = icolor
Line19.BorderColor = icolor
Line48.BorderColor = icolor
End Sub

Sub RD_Ram_color1()
Line15.BorderColor = QBColor(12)
Line16.BorderColor = QBColor(12)
Line17.BorderColor = QBColor(12)
Line18.BorderColor = QBColor(12)
Line19.BorderColor = QBColor(12)
Line20.BorderColor = QBColor(12)
Line21.BorderColor = QBColor(12)
Line48.BorderColor = QBColor(12)
End Sub

Sub RD_Ram_QBcolor1()
Line15.BorderColor = icolor
Line16.BorderColor = icolor
Line17.BorderColor = icolor
Line18.BorderColor = icolor
Line19.BorderColor = icolor
Line20.BorderColor = icolor
Line21.BorderColor = icolor
Line48.BorderColor = icolor
End Sub

Sub WR_Latch377_color1()
Line8.BorderColor = QBColor(12)
Line9.BorderColor = QBColor(12)
Line10.BorderColor = QBColor(12)
Line11.BorderColor = QBColor(12)
Line12.BorderColor = QBColor(12)
Line51.BorderColor = QBColor(12)
End Sub

Sub WR_Latch377_QBcolor1()
Line8.BorderColor = icolor
Line9.BorderColor = icolor
Line10.BorderColor = icolor
Line11.BorderColor = icolor
Line12.BorderColor = icolor
Line51.BorderColor = icolor
End Sub

Sub WR_NoAdd_color1()
Line8.BorderColor = QBColor(12)
Line9.BorderColor = QBColor(12)
Line10.BorderColor = QBColor(12)
Line11.BorderColor = QBColor(12)
```

```
Line12.BorderColor = QBColor(12)
Line49.BorderColor = QBColor(12)
End Sub

Sub WR_NoAdd_QBcolor1()
Line8.BorderColor = icolor
Line9.BorderColor = icolor
Line10.BorderColor = icolor
Line11.BorderColor = icolor
Line12.BorderColor = icolor
Line49.BorderColor = icolor
End Sub

Sub WR_Ram_color1()
Line8.BorderColor = QBColor(12)
Line9.BorderColor = QBColor(12)
Line10.BorderColor = QBColor(12)
Line11.BorderColor = QBColor(12)
Line12.BorderColor = QBColor(12)
Line13.BorderColor = QBColor(12)
Line14.BorderColor = QBColor(12)
Line49.BorderColor = QBColor(12)
End Sub

Sub WR_Ram_QBcolor1()
Line8.BorderColor = icolor
Line9.BorderColor = icolor
Line10.BorderColor = icolor
Line11.BorderColor = icolor
Line12.BorderColor = icolor
Line13.BorderColor = icolor
Line14.BorderColor = icolor
Line49.BorderColor = icolor
End Sub

Private Sub cmdINT0_Click()
iMemoriaRAMint(136, 6) = 1  'Activamos la bandera de interrupción
                            'externa IE0
ActualizacionRAM
Form2.Atencion_Interrupciones
End Sub

Private Sub cmdINT1_Click()
iMemoriaRAMint(136, 4) = 1  'Activamos la bandera de interrupción
                            'externa IE1
ActualizacionRAM
Form2.Atencion_Interrupciones
End Sub

Private Sub cmdT0_Click()
If ejec1 > 0 Then
   If Val(sInstruccionesOperandos(ejec1 - 1, 9)) = 1 Then
      If Val(sInstruccionesOperandos(ejec1, 9)) <> 1 Then
         ETimer0 = True
         Form2.Temp_Cont_0
         ETimer0 = False
         Form2.Atencion_Interrupciones
      ElseIf Val(sInstruccionesOperandos(ejec1, 9)) = 1 Then
         If STimer0 Then 'Si conto en el ciclo de la instruccion anterior
            STimer0 = False
         Else
            STimer0 = True
            ETimer0 = True
            Form2.Temp_Cont_0
            ETimer0 = False
            Form2.Atencion_Interrupciones
         End If
      End If
   Else
      If Val(sInstruccionesOperandos(ejec1, 9)) = 1 Then
         STimer0 = True
```

```
        End If
        ETimer0 = True
        Form2.Temp_Cont_0
        ETimer0 = False
        Form2.Atencion_Interrupciones
    End If
ElseIf ejec1 = 0 And Val(sInstruccionesOperandos(ejec1, 9)) <> 1 Then
        ETimer0 = True
        Form2.Temp_Cont_0
        ETimer0 = False
        Form2.Atencion_Interrupciones
End If
End Sub

Private Sub cmdT1_Click()
If ejec1 > 0 Then
    If Val(sInstruccionesOperandos(ejec1 - 1, 9)) = 1 Then
        If Val(sInstruccionesOperandos(ejec1, 9)) <> 1 Then
            ETimer1 = True
            Form2.Temp_Cont_1
            ETimer1 = False
            Form2.Atencion_Interrupciones
        ElseIf Val(sInstruccionesOperandos(ejec1, 9)) = 1 Then
            If STimer1 Then 'Si conto en el ciclo de la instruccion anterior
                STimer1 = False
            Else
                STimer1 = True
                ETimer1 = True
                Form2.Temp_Cont_1
                ETimer1 = False
                Form2.Atencion_Interrupciones
            End If
        End If
    Else
        If Val(sInstruccionesOperandos(ejec1, 9)) = 1 Then
            STimer1 = True
        End If
        ETimer1 = True
        Form2.Temp_Cont_1
        ETimer1 = False
        Form2.Atencion_Interrupciones
    End If
ElseIf ejec1 = 0 And Val(sInstruccionesOperandos(ejec1, 9)) <> 1 Then
        ETimer1 = True
        Form2.Temp_Cont_1
        ETimer1 = False
        Form2.Atencion_Interrupciones
End If
End Sub

Private Sub Image9_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)
If Button = 2 Then  'Sólo para el boton derecho
    iBuffer_P1 = True
    Form2.mnuDatosP1Buffer_D.Caption = iValor_In_P1 & " D"
    iX = iValor_In_P1
    ConversionD_H
    Form2.mnuDatosP1Buffer_H.Caption = sValorH & " H"
    Form2.mnuSwitcharP1.Caption = "Configurar Interruptores"
    PopupMenu Form2.mnuPuertoP1, vbPopupMenuLeftAlign
End If
End Sub

Private Sub img2KRam_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)
If Button = 2 Then  'Sólo para el boton derecho
    PopupMenu Form2.mnuMemoriasRAM, vbPopupMenuLeftAlign
End If
End Sub

Private Sub Img2KRom_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)
If Button = 2 Then  'Sólo para el boton derecho
    PopupMenu Form2.mnuMemoriasROM, vbPopupMenuLeftAlign
```

```
End If
End Sub

Private Sub imgBuffer_Bits_Click(Index As Integer)
If imgBuffer_Bits(Index).Tag = 1 Then
   imgBuffer_Bits(Index).Picture = frmImagenes.imgP1_01.Picture  'LoadPicture("c:\8031\dibpp\p1_01.bmp")
   imgBuffer_Bits(Index).Tag = "0"
Else
   imgBuffer_Bits(Index).Picture = frmImagenes.imgP1_11.Picture  'LoadPicture("c:\8031\dibpp\p1_11.bmp")
   imgBuffer_Bits(Index).Tag = "1"
End If
For ip1in = 0 To 7
   saValor_InB_Buffer(ip1in) = Val(imgBuffer_Bits(7 - ip1in).Tag)
Next ip1in
For ip1in = 0 To 7
   iValorB(ip1in) = saValor_InB_Buffer(ip1in)
Next ip1in
ConversionB_D
iValor_In_Buffer = iValorD
iX = iValorD
ConversionD_H
sValor_InH_Buffer = sValorH
End Sub

Private Sub imgBuffer244_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)
If Button = 2 Then  'Sólo para el boton derecho
   PopupMenu Form2.mnuBuffer541, vbPopupMenuLeftAlign
End If
End Sub

Private Sub imgDecodificador138_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)
If Button = 2 Then  'Sólo para el boton derecho
   PopupMenu Form2.mnuDireccionamiento, vbPopupMenuLeftAlign
End If
End Sub

Private Sub imgLatch_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)
If Button = 2 Then  'Sólo para el boton derecho
   PopupMenu Form2.mnuLatchz, vbPopupMenuLeftAlign
End If
End Sub

Private Sub imgLatch373_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)
For i377 = 0 To 7
   iValorB(i377) = saValor_Latch377(i377)
Next i377
ConversionB_D
Form2.mnuDatosLatch377_D.Caption = iValorD & " D"
iX = iValorD
ConversionD_H
Form2.mnuDatosLatch377_H.Caption = sValorH & " H"
If Button = 2 Then  'Sólo para el boton derecho
   PopupMenu Form2.mnuLatch377, vbPopupMenuLeftAlign
End If
End Sub

Private Sub imgMicro_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)
If Button = 2 Then  'Sólo para el boton derecho
   PopupMenu Form2.mnuMicro, vbPopupMenuLeftAlign
End If
End Sub

Private Sub imgPuerto_P1_Click(Index As Integer)
If Index <= 3 Then
   If imgPuerto_P1(Index).Tag = "1" Then
      imgPuerto_P1(Index).Picture = frmImagenes.imgP1_01.Picture
      imgPuerto_P1(Index).Tag = "0"
   Else
      imgPuerto_P1(Index).Picture = frmImagenes.imgP1_11.Picture
      imgPuerto_P1(Index).Tag = "1"
   End If
```

```
    For ip1in = 4 To 7
        saValor_InB_P1(ip1in) = Val(imgPuerto_P1(7 - ip1in).Tag)
    Next ip1in
    For ip1in = 0 To 7
        iValorB(ip1in) = saValor_InB_P1(ip1in)
    Next ip1in
    ConversionB_D
    iValor_In_P1 = iValorD
    iX = iValorD
    ConversionD_H
    sValor_InH_P1 = sValorH
End If
End Sub

Private Sub imgSwitch_Buf_MouseDown(Button As Integer, Shift As integer, X As Single, Y As Single)
If Button = 2 Then  'Sólo para el boton derecho
    iBuffer_P1 = False
    Form2.mnuDatosP1Buffer_D.Caption = iValor_In_Buffer & " D"
    iX = iValor_In_Buffer
    ConversionD_H
    Form2.mnuDatosP1Buffer_H.Caption = sValorH & " H"
    Form2.mnuSwitcharP1.Caption = "Configurar Interruptores"
    PopupMenu Form2.mnuPuertoP1, vbPopupMenuLeftAlign
End If
End Sub

Private Sub lblReset_Click()
Respuesta = MsgBox("Seguro que desea Resetear?", vbOKCancel + vbQuestion, "RESET")
If Respuesta = vbOK Then
    Step = 0
    ejec1 = -1
    iMemoriaRAMintD(224) = 0
    iMemoriaRAMintD(240) = 0
    iMemoriaRAMintD(208) = 0
    iMemoriaRAMintD(129) = 47
    iMemoriaRAMintD(130) = 0
    iMemoriaRAMintD(131) = 0
    iMemoriaRAMintD(128) = 255
    iMemoriaRAMintD(144) = 255
    iMemoriaRAMintD(160) = 255
    iMemoriaRAMintD(176) = 255
    iMemoriaRAMintD(184) = 0
    iMemoriaRAMintD(168) = 0
    iMemoriaRAMintD(137) = 0
    iMemoriaRAMintD(136) = 0
    iMemoriaRAMintD(140) = 0
    iMemoriaRAMintD(138) = 0
    iMemoriaRAMintD(141) = 0
    iMemoriaRAMintD(139) = 0
    iMemoriaRAMintD(152) = 0
    iMemoriaRAMintD(153) = 0
    iMemoriaRAMintD(135) = 0
    ActualizacionRAMD_BH
    fraTIMER0.Visible = False
    lblTIMER0.Visible = False
    fraTIMER1.Visible = False
    lblTIMER1.Visible = False
    Label14.Visible = False
    lblDec_Salida.Visible = False
    lblDecoder.Visible = False
    Line53.Visible = False
    Line60.Visible = False
    STimer0 = False
    STimer1 = False
End If
End Sub
```

frmDirec_Ram

```
'Declaración de variables
Dim iRangoRam  As Byte
Dim iRangoRam1  As Byte

Sub RangoRam()
Select Case iRangoRam
   Case 1
      Label10(iRangoRam1).Caption = "8192"
      Label10(1 + iRangoRam1).Caption = "16383"
      Label10(2 + iRangoRam1).Caption = "2000"
      Label10(3 + iRangoRam1).Caption = "3FFF"
   Case 2
      Label10(iRangoRam1).Caption = "16384"
      Label10(1 + iRangoRam1).Caption = "24575"
      Label10(2 + iRangoRam1).Caption = "4000"
      Label10(3 + iRangoRam1).Caption = "5FFF"
   Case 3
      Label10(iRangoRam1).Caption = "24576"
      Label10(1 + iRangoRam1).Caption = "32767"
      Label10(2 + iRangoRam1).Caption = "6000"
      Label10(3 + iRangoRam1).Caption = "7FFF"
   Case 4
      Label10(iRangoRam1).Caption = "32768"
      Label10(1 + iRangoRam1).Caption = "40959"
      Label10(2 + iRangoRam1).Caption = "8000"
      Label10(3 + iRangoRam1).Caption = "9FFF"
   Case 5
      Label10(iRangoRam1).Caption = "40960"
      Label10(1 + iRangoRam1).Caption = "49151"
      Label10(2 + iRangoRam1).Caption = "A000"
      Label10(3 + iRangoRam1).Caption = "BFFF"
   Case 6
      Label10(iRangoRam1).Caption = "49152"
      Label10(1 + iRangoRam1).Caption = "57343"
      Label10(2 + iRangoRam1).Caption = "C000"
      Label10(3 + iRangoRam1).Caption = "DFFF"
   Case 7
      Label10(iRangoRam1).Caption = "57344"
      Label10(1 + iRangoRam1).Caption = "65535"
      Label10(2 + iRangoRam1).Caption = "E000"
      Label10(3 + iRangoRam1).Caption = "FFFF"
End Select
End Sub

Private Sub cmdDirec_Ram_Click()
iRam = 0
iBuffer = 0
iLatch = 0
For iRBL = 0 To 2
   iRam = iRam + Val(Mid(Label7(iRBL), 2, 1)) * 2 ^ (2 - iRBL)
Next iRBL
For iRBL = 0 To 2
   iBuffer = iBuffer + Val(Mid(Label7(3 + iRBL), 2, 1)) * 2 ^ (2 - iRBL)
Next iRBL
For iRBL = 0 To 2
   iLatch = iLatch + Val(Mid(Label7(6 + iRBL), 2, 1)) * 2 ^ (2 - iRBL)
Next iRBL
iRangoRam = iRam
iRangoRam1 = 0
RangoRam
iRangoRam = iBuffer
iRangoRam1 = 4
RangoRam
iRangoRam = iLatch
iRangoRam1 = 8
RangoRam
If iRam = iBuffer Then
   MsgBox "No es posible usar la misma Decodificación", vbOKOnly + vbCritical, "Decodificación"
ElseIf iRam = iLatch Then
```

```
    MsgBox "No es posible usar la misma Decodificación", vbOKOnly + vbCritical, "Decodificación"
  ElseIf iBuffer = iLatch Then
    MsgBox "No es posible usar la misma Decodificación", vbOKOnly + vbCritical, "Decodificación"
  Else
    Unload frmDirec_Ram
  End If
  End Sub

  Private Sub Form_Load()
  iX1d_b = iRam
  ConversionD_B
  For iRBL = 0 To 2
    If iMatrizB(5 + iRBL) = 0 Then
      Label7(iRBL).Caption = " 0 "
      Label7(iRBL).BackColor = QBColor(15)
    Else
      Label7(iRBL).Caption = " 1 "
      Label7(iRBL).BackColor = QBColor(12)
    End If
  Next iRBL
  iRangoRam = iRam
  iRangoRam1 = 0
  RangoRam
  iX1d_b = iBuffer
  ConversionD_B
  For iRBL = 0 To 2
    If iMatrizB(5 + iRBL) = 0 Then
      Label7(3 + iRBL).Caption = " 0 "
      Label7(3 + iRBL).BackColor = QBColor(15)
    Else
      Label7(3 + iRBL).Caption = " 1 "
      Label7(3 + iRBL).BackColor = QBColor(12)
    End If
  Next iRBL
  iRangoRam = iBuffer
  iRangoRam1 = 4
  RangoRam
  iX1d_b = iLatch
  ConversionD_B
  For iRBL = 0 To 2
    If iMatrizB(5 + iRBL) = 0 Then
      Label7(6 + iRBL).Caption = " 0 "
      Label7(6 + iRBL).BackColor = QBColor(15)
    Else
      Label7(6 + iRBL).Caption = " 1 "
      Label7(6 + iRBL).BackColor = QBColor(12)
    End If
  Next iRBL
  iRangoRam = iLatch
  iRangoRam1 = 8
  RangoRam
  End Sub

  Private Sub Label7_Click(Index As Integer)
  If Label7(Index).Caption = " 0 " Then
    Label7(Index).Caption = " 1 "
    Label7(Index).BackColor = QBColor(12)
  Else
    Label7(Index).Caption = " 0 "
    Label7(Index).BackColor = QBColor(15)
  End If
  iRam = 0
  iBuffer = 0
  iLatch = 0
  For iRBL = 0 To 2
    iRam = iRam + Val(Mid(Label7(iRBL), 2, 1)) * 2 ^ (2 - iRBL)
  Next iRBL
  For iRBL = 0 To 2
    iBuffer = iBuffer + Val(Mid(Label7(3 + iRBL), 2, 1)) * 2 ^ (2 - iRBL)
  Next iRBL
  For iRBL = 0 To 2
```

```
    iLatch = iLatch + Val(Mid(Label7(6 + iRBL), 2, 1)) * 2 ^ (2 - iRBL)
  Next iRBL
  If iRam = iBuffer Then
    MsgBox "No es posible usar la misma Decodificación", vbOKOnly + vbCritical, "Decodificación"
  ElseIf iRam = iLatch Then
    MsgBox "No es posible usar la misma Decodificación", vbOKOnly + vbCritical, "Decodificación"
  ElseIf iBuffer = iLatch Then
    MsgBox "No es posible usar la misma Decodificación", vbOKOnly + vbCritical, "Decodificación"
  End If
  iRangoRam = iRam
  iRangoRam1 = 0
  RangoRam
  iRangoRam = iBuffer
  iRangoRam1 = 4
  RangoRam
  iRangoRam = iLatch
  iRangoRam1 = 8
  RangoRam
  End Sub
```

frmEdit_Ins

```
'Declaración de variables
Dim txt1O As Boolean
Dim txt2O As Boolean
Dim txt3O As Boolean
Dim txtdbc1O As Boolean
Dim txtdbc2O As Boolean

Sub CodigoInstruccion_Texto()
'Permite formar el código completo de la instrucción
If txt1O And txt2O And txt3O Then
   If txtdbc1O Then
      txtCodigoInstruccion.Text = dbcOpcodeIns.Text & "    " & dbcPrimerOperando.Text & "," &
            txtSegundoOperando.Text & "," & txtTercerOperando.Text
   Else
      txtCodigoInstruccion.Text = dbcOpcodeIns.Text & "    " & txtPrimerOperando.Text & "," &
            txtSegundoOperando.Text & "," & txtTercerOperando.Text
   End If
   If txtdbc2O Then
      txtCodigoInstruccion.Text = dbcOpcodeIns.Text & "    " & txtPrimerOperando.Text & "," &
            dbcSegundoOperando.Text & "," & txtTercerOperando.Text
   ElseIf txtdbc2O = False And txtdbc1O = False Then
      txtCodigoInstruccion.Text = dbcOpcodeIns.Text & "    " & txtPrimerOperando.Text & "," &
            txtSegundoOperando.Text & "," & txtTercerOperando.Text
   End If
ElseIf txt1O And txt2O Then
   If txtdbc1O Then
      txtCodigoInstruccion.Text = dbcOpcodeIns.Text & "    " & dbcPrimerOperando.Text & "," &
            txtSegundoOperando.Text
   Else
      txtCodigoInstruccion.Text = dbcOpcodeIns.Text & "    " & txtPrimerOperando.Text & "," &
            txtSegundoOperando.Text
   End If
   If txtdbc2O Then
      txtCodigoInstruccion.Text = dbcOpcodeIns.Text & "    " & txtPrimerOperando.Text & "," &
            dbcSegundoOperando.Text
   ElseIf txtdbc2O = False And txtdbc1O = False Then
      txtCodigoInstruccion.Text = dbcOpcodeIns.Text & "    " & txtPrimerOperando.Text & "," &
            txtSegundoOperando.Text
   End If
ElseIf txt1O Then
   If txtdbc1O Then
      txtCodigoInstruccion.Text = dbcOpcodeIns.Text & "    " & dbcPrimerOperando.Text
   Else
      txtCodigoInstruccion.Text = dbcOpcodeIns.Text & "    " & txtPrimerOperando.Text
   End If
Else
   txtCodigoInstruccion.Text = dbcOpcodeIns.Text
End If
txtCodigoInstruccion.Text = txtEtiqueta.Text & "    " & txtCodigoInstruccion.Text
```

```
End Sub

Private Sub cmdAceptar_Click()
'Utilizamos el menu Save As de Windows
'para cargar los programas
On Error GoTo Cancel1
CommonDialog1.CancelError = True
CommonDialog1.Filter = "Cargar Programa (*.asm)|*.asm|Cargar Programa (*.txt)|*.txt|Todos (*.*)|*.*"
CommonDialog1.FilterIndex = 1
CommonDialog1.InitDir = "c:\8031\ArUsuario"
CommonDialog1.Action = 2
sFileName1 = CommonDialog1.filename
If sFileName1 = "" Then
Cancel1:
Exit Sub
End If
FileNumber1 = FreeFile
Open sFileName1 For Output As #FileNumber1
For iedit = 0 To iLineasEntrada
   Print #FileNumber1, saMatrizInstrucciones(iedit)
Next iedit
Close #FileNumber1
Unload Me
End Sub

Private Sub cmdAplicar_Click()
iLineasEntrada = iLineasEntrada + 1
saMatrizInstrucciones(iLineasEntrada) = txtCodigoInstruccion
dbcInstrucciones1.AddItem txtCodigoInstruccion
dbcInstrucciones1.ListIndex = iLineasEntrada
txtEtiqueta.Text = ""
End Sub

Private Sub cmdCancelar_Click()
Unload Me
End Sub

Private Sub dbcOpcodeIns_Change()
Dim sConsulta As String
txtPrimerOperando.Visible = True
txtPrimerOperando.Enabled = True
dbcPrimerOperando.Visible = False
dbcPrimerOperando.Enabled = False
txtdbc1O = False
txtSegundoOperando.Visible = True
txtSegundoOperando.Enabled = True
dbcSegundoOperando.Visible = False
dbcSegundoOperando.Enabled = False
txtdbc2O = False
sConsulta = "Select PrimerOperando, SegundoOperando, TercerOperando, Operandos123 From
          Operandos Where Opcode = '" & dbcOpcodeIns & "'"
Data2.RecordSource = sConsulta
Data2.Refresh
If Data2.Recordset("Operandos123") <> "" Then
   dbcOperandos123.Text = Data2.Recordset("Operandos123")
Else
   dbcOperandos123.Text = ""
End If
If Data2.Recordset("PrimerOperando") <> "" Then
   txtPrimerOperando.Text = Data2.Recordset("PrimerOperando")
   txt1O = True
Else
   txtPrimerOperando.Text = ""
   txt1O = False
End If
If Data2.Recordset("SegundoOperando") <> "" Then
   txtSegundoOperando.Text = Data2.Recordset("SegundoOperando")
   txt2O = True
Else
   txtSegundoOperando.Text = ""
   txt2O = False
```

```
End If
If Data2.Recordset("TercerOperando") <> "" Then
   txtTercerOperando.Text = Data2.Recordset("TercerOperando")
   txt3O = True
Else
   txtTercerOperando.Text = ""
   txt3O = False
End If
CodigoInstruccion_Texto
End Sub

Private Sub dbcOperandos123_Click(Area As Integer)
Dim sConsulta1 As String
Dim sConsulta2 As String
sConsulta1 = "Select PrimerOperando, SegundoOperando, TercerOperando From Operandos Where
                    Operandos123 = '" & dbcOperandos123 & "'"
Data3.RecordSource = sConsulta1
Data3.Refresh
If Data3.Recordset("PrimerOperando") <> "" Then
   txtPrimerOperando.Text = Data3.Recordset("PrimerOperando")
   txt1O = True
   If txtPrimerOperando.Text = "@Ri" Or txtPrimerOperando = "Rn" Then
      txtPrimerOperando.Visible = False
      txtPrimerOperando.Enabled = False
      dbcPrimerOperando.Visible = True
      dbcPrimerOperando.Enabled = True
      sConsulta2 = "Select VariantesRiRn From VariantesRiRn Where RiRn = '" &
           txtPrimerOperando.Text & "'"
      Data4.RecordSource = sConsulta2
      Data4.Refresh
      dbcPrimerOperando.Text = Data4.Recordset("VariantesRiRn")
      txtdbc1O = True
      txtdbc2O = False
   Else
      txtPrimerOperando.Visible = True
      txtPrimerOperando.Enabled = True
      dbcPrimerOperando.Visible = False
      dbcPrimerOperando.Enabled = False
      txtdbc1O = False
   End If
Else
   txtPrimerOperando = ""
   txt1O = False
End If
If Data3.Recordset("SegundoOperando") <> "" Then
   txtSegundoOperando.Text = Data3.Recordset("SegundoOperando")
   txt2O = True
   If txtSegundoOperando.Text = "@Ri" Or txtSegundoOperando = "Rn" Then
      txtSegundoOperando.Visible = False
      txtSegundoOperando.Enabled = False
      dbcSegundoOperando.Visible = True
      dbcSegundoOperando.Enabled = True
      sConsulta2 = "Select VariantesRiRn From VariantesRiRn Where RiRn = '" &
           txtSegundoOperando.Text & "'"
      Data4.RecordSource = sConsulta2
      Data4.Refresh
      dbcSegundoOperando.Text = Data4.Recordset("VariantesRiRn")
      txtdbc2O = True
      txtdbc1O = False
   Else
      txtSegundoOperando.Visible = True
      txtSegundoOperando.Enabled = True
      dbcSegundoOperando.Visible = False
      dbcSegundoOperando.Enabled = False
      txtdbc2O = False
   End If
Else
   txtSegundoOperando.Text = ""
   txt2O = False
End If
If Data3.Recordset("TercerOperando") <> "" Then
```

```
    txtTercerOperando.Text = Data2.Recordset("TercerOperando")
    txt3O = True
  Else
    txtTercerOperando.Text = ""
    txt3O = False
  End If
  CodigoInstruccion_Texto
  End Sub

  Private Sub dbcPrimerOperando_Change()
  CodigoInstruccion_Texto
  End Sub

  Private Sub dbcSegundoOperando_Change()
  CodigoInstruccion_Texto
  End Sub

  Private Sub Form_Activate()
  dbcOpcodeIns.Text = Data1.Recordset("Opcode")
  dbcOperandos123.Text = Data2.Recordset("Operandos123")
  txtPrimerOperando.Text = Data2.Recordset("PrimerOperando")
  If Data2.Recordset("SegundoOperando") <> "" Then
    txtSegundoOperando.Text = Data2.Recordset("SegundoOperando")
  Else
    txtSegundoOperando.Text = ""
  End If
  If Data2.Recordset("TercerOperando") <> "" Then
    txtTercerOperando.Text = Data2.Recordset("TercerOperando")
  Else
    txtTercerOperando.Text = ""
  End If
  txtCodigoInstruccion.Text = dbcOpcodeIns.Text & "     " & txtPrimerOperando.Text '& "," &
            txtSegundoOperando.Text & "," & txtTercerOperando.Text
  End Sub

  Private Sub Form_Load()
  iLineasEntrada = -1
  End Sub

  Private Sub txtEtiqueta_Change()
  CodigoInstruccion_Texto
  End Sub

  Private Sub txtPrimerOperando_Change()
  CodigoInstruccion_Texto
  End Sub

  Private Sub txtSegundoOperando_Change()
  CodigoInstruccion_Texto
  End Sub

  Private Sub txtTercerOperando_Change()
  CodigoInstruccion_Texto
  End Sub
```

frmMemoriaROM

```
  'Declaració de variables
  Dim saMatrizMemorias(9) As String
  Dim iI As Integer
  Dim iJ As Integer
  Dim iBit As Integer

  Private Sub Command1_Click()
  grdRom.Rows = Val(sInstruccionesOperandos(iMatriz, 25)) + 6
  For iI = 1 To 9
    grdRom.ColAlignment(iI) = 2
    grdRom.ColWidth(iI) = 250
  Next
  grdRom.FixedAlignment(0) = 2
  grdRom.Row = 0
```

```
 For iI = 0 To 8
    grdRom.Col = iI
    grdRom.Text = saMatrizMemorias(iI + 1)
 Next
 grdRom.Col = 9
 grdRom.Text = " H"
 For iI = 1 To Val(sInstruccionesOperandos(iMatriz, 25)) + 5
    grdRom.Row = iI
    For iJ = 1 To 8
       grdRom.Col = iJ
       grdRom.Text = iMemoriaROMext(iI - 1, iJ - 1)
       iValorB(iJ - 1) = iMemoriaROMext(iI - 1, iJ - 1)
    Next iJ
    ConversionB_D
    iX = iValorD
    ConversionD_H
    grdRom.Col = 9
    grdRom.Text = sValorH
 Next iI
 grdRom.Col = 0
 For iJ = 0 To Val(sInstruccionesOperandos(iMatriz, 25)) + 4
    iX = iJ
    grdRom.Row = iJ + 1
    ConversionDPTRD_H
    grdRom.Text = sValor16BitsH
 Next
 End Sub

 Private Sub Form_Load()
 saMatrizMemorias(1) = "addrs"
 saMatrizMemorias(2) = "b7"
 saMatrizMemorias(3) = "b6"
 saMatrizMemorias(4) = "b5"
 saMatrizMemorias(5) = "b4"
 saMatrizMemorias(6) = "b3"
 saMatrizMemorias(7) = "b2"
 saMatrizMemorias(8) = "b1"
 saMatrizMemorias(9) = "b0"
 grdRom.Rows = Val(sInstruccionesOperandos(iMatriz, 25)) + 6
 For iI = 1 To 9
    grdRom.ColAlignment(iI) = 2
    grdRom.ColWidth(iI) = 250
 Next
 grdRom.FixedAlignment(0) = 2
 grdRom.Row = 0
 For iI = 0 To 8
    grdRom.Col = iI
    grdRom.Text = saMatrizMemorias(iI + 1)
 Next
 grdRom.Col = 9
 grdRom.Text = " H"
 For iI = 1 To Val(sInstruccionesOperandos(iMatriz, 25)) + 5
    grdRom.Row = iI
    For iJ = 1 To 8
       grdRom.Col = iJ
       grdRom.Text = iMemoriaROMext(iI - 1, iJ - 1)
       iValorB(iJ - 1) = iMemoriaROMext(iI - 1, iJ - 1)
    Next iJ
    ConversionB_D
    iX = iValorD
    ConversionD_H
    grdRom.Col = 9
    grdRom.Text = sValorH
 Next iI
 grdRom.Col = 0
 For iJ = 0 To Val(sInstruccionesOperandos(iMatriz, 25)) + 4
    iX = iJ
    grdRom.Row = iJ + 1
    ConversionDPTRD_H
    grdRom.Text = sValor16BitsH
 Next
```

```
End Sub
```

frmMemorias

```
'Declaración de variables
Dim saMatrizMemorias(9) As String
Dim iI As Integer
Dim iJ As Integer
Dim iBit As Integer

Private Sub cmdMemorias_Click()
ActualizacionRAM
For iI = 1 To 256
   grdMemorias.Row = iI
   For iJ = 1 To 8
      grdMemorias.Col = iJ
      grdMemorias.Text = iMemoriaRAMint(iI - 1, iJ - 1) 'Aqui se escibiran los datos que se guarden en una matriz
   Next
   grdMemorias.Col = 9
   grdMemorias.Text = sMemoriaRAMintH(iI - 1)
Next
End Sub

Private Sub Form_Load()
saMatrizMemorias(1) = "addrs"
saMatrizMemorias(2) = "b7"
saMatrizMemorias(3) = "b6"
saMatrizMemorias(4) = "b5"
saMatrizMemorias(5) = "b4"
saMatrizMemorias(6) = "b3"
saMatrizMemorias(7) = "b2"
saMatrizMemorias(8) = "b1"
saMatrizMemorias(9) = "b0"
For iI = 1 To 9
   grdMemorias.ColAlignment(iI) = 2
   grdMemorias.ColWidth(iI) = 250
Next
grdMemorias.FixedAlignment(0) = 2
grdMemorias.Row = 0
For iI = 0 To 8
   grdMemorias.Col = iI
   grdMemorias.Text = saMatrizMemorias(iI + 1)
Next
For iI = 1 To 256
   grdMemorias.Row = iI
   For iJ = 1 To 8
      grdMemorias.Col = iJ
      grdMemorias.Text = iMemoriaRAMint(iI - 1, iJ - 1) 'Aqui se escibiran los datos que se guarden en
                                                        'una matriz
   Next
   grdMemorias.Col = 9
   grdMemorias.Text = sMemoriaRAMintH(iI - 1)
Next
grdMemorias.Col = 9
grdMemorias.Row = 130 + 1
grdMemorias.Text = Mid(sMemoriaRAMintH(130), 3, 2)
grdMemorias.Row = 131 + 1
grdMemorias.Text = Mid(sMemoriaRAMintH(130), 1, 2)
grdMemorias.Col = 0
For iJ = 0 To 255
   iX = iJ
   grdMemorias.Row = iJ + 1
   ConversionD_H
   grdMemorias.Text = sValorH & "H"
Next
grdMemorias.Col = 9
grdMemorias.Row = 0
grdMemorias.Text = " H"
grdMemorias.ColAlignment(9) = 2
grdMemorias.ColAlignment(10) = 2
grdMemorias.ColAlignment(11) = 2
```

```
grdMemorias.ColWidth(10) = 900
grdMemorias.ColWidth(11) = 400
grdMemorias.Col = 10
For iI = 0 To 3
   For iJ = 1 To 8
      grdMemorias.Row = iJ + iI * 8
      grdMemorias.Text = "Banco " & iI
   Next iJ
Next iI
grdMemorias.Col = 11
For iI = 0 To 3
   For iJ = 0 To 7
      grdMemorias.Row = iJ + iI * 8 + 1
      grdMemorias.Text = "R" & iJ
   Next iJ
Next iI
grdMemorias.Col = 10
For iI = 33 To 48
   grdMemorias.Row = iI
   grdMemorias.Text = "*DATOS"
Next iI
For iI = 49 To 128
   grdMemorias.Row = iI
   grdMemorias.Text = "DATOS"
Next iI
grdMemorias.Col = 10
grdMemorias.Row = 129
grdMemorias.Text = "*P0"
grdMemorias.Row = 130
grdMemorias.Text = "SP"
grdMemorias.Row = 131
grdMemorias.Text = "DPL"
grdMemorias.Row = 132
grdMemorias.Text = "DPH"
grdMemorias.Row = 136
grdMemorias.Text = "PCON"
grdMemorias.Row = 137
grdMemorias.Text = "*TCON"
grdMemorias.Row = 138
grdMemorias.Text = "TMOD"
grdMemorias.Row = 139
grdMemorias.Text = "TL0"
grdMemorias.Row = 140
grdMemorias.Text = "TL1"
grdMemorias.Row = 141
grdMemorias.Text = "TH0"
grdMemorias.Row = 142
grdMemorias.Text = "TH1"
grdMemorias.Row = 145
grdMemorias.Text = "*P1"
grdMemorias.Row = 153
grdMemorias.Text = "*SCON"
grdMemorias.Row = 154
grdMemorias.Text = "SBUF"
grdMemorias.Row = 161
grdMemorias.Text = "*P2"
grdMemorias.Row = 169
grdMemorias.Text = "*IE"
grdMemorias.Row = 177
grdMemorias.Text = "*P3"
grdMemorias.Row = 185
grdMemorias.Text = "*IP"
grdMemorias.Row = 209
grdMemorias.Text = "*PSW"
grdMemorias.Row = 225
grdMemorias.Text = "*ACC"
grdMemorias.Row = 241
grdMemorias.Text = "*B"
'Pintamos la localidad de memoria que esta apuntando
'el Stack Pointer
grdMemorias.Row = SP + 1
```

```
grdMemorias.Text = grdMemorias.Text & " SP"
End Sub

Private Sub grdMemorias_Click()
If grdMemorias.Col >= 1 And grdMemorias.Col <= 8 Then
   If grdMemorias.Row >= 1 And grdMemorias.Row <= 256 Then
      iBit = grdMemorias.Text
      If iBit = 0 Then
         grdMemorias.Text = 1
         iMemoriaRAMint(grdMemorias.Row - 1, grdMemorias.Col - 1) = 1
      Else
         grdMemorias.Text = 0
         iMemoriaRAMint(grdMemorias.Row - 1, grdMemorias.Col - 1) = 0
      End If
   Else
      Exit Sub
   End If
Else
   Exit Sub
End If
End Sub
```

frmRamExterna

```
'Declaración de variables
Dim saMatrizMemorias(9) As String

Private Sub cmdAceptarRamExterna_Click()
Dim iDE As Integer
Dim iA As Integer
Dim iNumfilas As Integer
LineaLeida1 = "#" & txtDE.Text
If LineaLeida1 = "#" Then
   MsgBox "No hay datos ingresados", vbOKOnly + vbExclamation, frmRamExterna.Caption
Else
   Y = 0
   Form2.Num_Data16
   If iNum_Data16 > 4095 Then
      MsgBox "Valor máximo en casilleros es 4095 ", vbOKOnly + vbCritical, frmRamExterna.Caption
   Else
      iDE = iNum_Data16
   End If
End If
LineaLeida1 = "#" & txtA.Text
If LineaLeida1 = "#" Then
   MsgBox "No hay datos ingresados", vbOKOnly + vbExclamation, frmRamExterna.Caption
Else
   Y = 0
   Form2.Num_Data16
   If iNum_Data16 > 4095 Then
      MsgBox "Valor máximo en casilleros es 4095", vbOKOnly + vbCritical, frmRamExterna.Caption
   Else
      iA = iNum_Data16
   End If
End If
If iA - iDE > 200 Then
   MsgBox "El Rango a visualizar debe ser memor a 200", vbOKOnly + vbExclamation, frmRamExterna.Caption
ElseIf iA - id < 0 Then
   MsgBox "El Rango es negativo", vbOKOnly + vbCritical, frmRamExterna.Caption
ElseIf iA - iDE < 20 Then
   iNumfilas = 20
   grdRamExterna.Rows = iNumfilas + 2
Else
   iNumfilas = iA - iDE
   grdRamExterna.Rows = iNumfilas + 2
End If
If KbytesRam = 1024 And (iDE > 1023 Or iA > 1023) Then
   MsgBox "Memoria Ram externa solo es de 1 Kbytes", vbOKOnly + vbCritical, frmRamExterna.Caption
ElseIf KbytesRam = 2048 And (iDE > 2047 Or iA > 2047) Then
   MsgBox "Memoria Ram externa solo es de 2 Kytes", vbOKOnly + vbCritical, frmRamExterna.Caption
ElseIf KbytesRam = 4096 And (iDE > 2047 Or iA > 2047) Then
```

```
    MsgBox "Memoria Ram externa solo es de 4 Kbytes", vbOKOnly + vbCritical, frmRamExterna.Caption
  End If
  For irext = 0 To iNumfilas
    grdRamExterna.Row = irext + 1
    For irext1 = 0 To 7
      grdRamExterna.Col = irext1 + 1
      grdRamExterna.Text = iMemoriaRAMext(iDE + irext, irext1)
      iValorB(irext1) = iMemoriaRAMext(iDE + irext, irext1)
    Next irext1
    ConversionB_D
    iX = iValorD
    ConversionD_H
    grdRamExterna.Col = 9
    grdRamExterna.Text = sValorH
  Next irext
  grdRamExterna.Col = 0
  For irext = 0 To iNumfilas
    iX = iDE + irext
    grdRamExterna.Row = irext + 1
    ConversionDPTRD_H
    grdRamExterna.Text = sValor16BitsH
  Next
  End Sub

  Private Sub Form_Load()
  saMatrizMemorias(1) = "addrs"
  saMatrizMemorias(2) = "b7"
  saMatrizMemorias(3) = "b6"
  saMatrizMemorias(4) = "b5"
  saMatrizMemorias(5) = "b4"
  saMatrizMemorias(6) = "b3"
  saMatrizMemorias(7) = "b2"
  saMatrizMemorias(8) = "b1"
  saMatrizMemorias(9) = "b0"
  If KbytesRam = 1024 Then
    frmRamExterna.Caption = "Memoria Ram Externa 1 Kbytes"
  ElseIf KbytesRam = 2048 Then
    frmRamExterna.Caption = "Memoria Ram Externa 2 Kbytes"
  ElseIf KbytesRam = 4096 Then
    frmRamExterna.Caption = "Memoria Ram Externa 4 Kbytes"
  End If
  For il = 1 To 9
    grdRamExterna.ColAlignment(il) = 2
    grdRamExterna.ColWidth(il) = 250
  Next
  grdRamExterna.FixedAlignment(0) = 2
  grdRamExterna.Row = 0
  For il = 0 To 8
    grdRamExterna.Col = il
    grdRamExterna.Text = saMatrizMemorias(il + 1)
  Next
  grdRamExterna.Col = 9
  grdRamExterna.Text = " H"
  For irext = 0 To 199
    grdRamExterna.Row = 1 + irext
    For irext1 = 0 To 7
      grdRamExterna.Col = 1 + irext1
      grdRamExterna.Text = iMemoriaRAMext(irext, irext1)
      iValorB(irext1) = iMemoriaRAMext(irext, irext1)
    Next irext1
    ConversionB_D
    iX = iValorD
    ConversionD_H
    grdRamExterna.Col = 9
    grdRamExterna.Text = sValorH
  Next irext
  grdRamExterna.Col = 0
  For irext = 0 To 199
    iX = irext
    grdRamExterna.Row = irext + 1
    ConversionDPTRD_H
```

```
    grdRamExterna.Text = sValor16BitsH
  Next
  End Sub
```

frmSwitcheo_P1

```
Private Sub cmdSwitcheo_P1_Aceptar_Click()
LineaLeida1 = txtSwitcheo_P1.Text
iLongitud1 = Len(LineaLeida1)
Y = 0
Form2.Num_Data
If iBuffer_P1 Then 'Si es en los interruptores de entrada a P1
   If iNum_Data <= 15 Then
      iValor_In_P1 = iNum_Data
      iX = iValor_In_P1
      ConversionD_H
      sValor_InH_P1 = sValorH
      iX1d_b = iValor_In_P1
      ConversionD_B
      For ip1in = 0 To 7
         saValor_InB_P1(ip1in) = iMatrizB(ip1in)
      Next ip1in
   Else
      MsgBox "El dato a ingresar máximo será 15", vbOKOnly + vbExclamation, "Configuración
                                                                    Interruptores"
   End If
Else
   iValor_In_Buffer = iNum_Data
   iX = iValor_In_Buffer
   ConversionD_H
   sValor_InH_Buffer = sValorH
   iX1d_b = iValor_In_Buffer
   ConversionD_B
   For ip1in = 0 To 7
      saValor_InB_Buffer(ip1in) = iMatrizB(ip1in)
   Next ip1in
End If
   If frmDiagrama1.Visible Then
      For ip1in = 0 To 7
      If saValor_InB_P1(ip1in) = 0 Then
         frmDiagrama1.imgPuerto_P1(7 - ip1in).Picture = frmImagenes.imgP1_01.Picture
      Else
         frmDiagrama1.imgPuerto_P1(7 - ip1in).Picture = frmImagenes.imgP1_11.Picture
      End If
      Next ip1in
   ElseIf frmDiagrama2.Visible Then
      If iBuffer_P1 Then
         For ip1in = 4 To 7
         If saValor_InB_P1(ip1in) = 0 Then
            frmDiagrama2.imgPuerto_P1(7 - ip1in).Picture = frmImagenes.imgP1_01.Picture
            frmDiagrama2.imgPuerto_P1(7 - ip1in).Tag = "0"
         Else
            frmDiagrama2.imgPuerto_P1(7 - ip1in).Picture = frmImagenes.imgP1_11.Picture
            frmDiagrama2.imgPuerto_P1(7 - ip1in).Tag = "1"
         End If
         Next ip1in
      Else
         For ip1in = 0 To 7
         If saValor_InB_Buffer(ip1in) = 0 Then
            frmDiagrama2.imgBuffer_Bits(7 - ip1in).Picture = frmImagenes.imgP1_01.Picture
            frmDiagrama2.imgBuffer_Bits(7 - ip1in).Tag = "0"
         Else
            frmDiagrama2.imgBuffer_Bits(7 - ip1in).Picture = frmImagenes.imgP1_11.Picture
            frmDiagrama2.imgBuffer_Bits(7 - ip1in).Tag = "1"
         End If
         Next ip1in
      End If
   End If
Unload Me
End Sub
```

```
Private Sub cmdSwitcheo_P1_Cancelar_Click()
Unload Me
End Sub
```

MDIForm1

```
Private Sub MDIForm_Load()
Form2.Show
Form2.Hide
Me.Top = -6
Me.Left = -6
Me.Height = 8600
Me.Width = 12040
frmDiagrama2.Top = 0
frmDiagrama2.Left = 0
frmDiagrama2.Height = 7450
frmDiagrama2.Width = 11870
End Sub

Private Sub mnum1_Click()
Form2.Smnu1
mnum1.Checked = True
mnum2.Checked = False
mnum3.Checked = False
mnum4.Checked = False
End Sub

Private Sub mnum1KbytesRAM_Click()
Form2.Smnu1KbytesRAM
mnum2KbytesRAM.Checked = False
mnum1KbytesRAM.Checked = True
mnum4KbytesRAM.Checked = False
End Sub

Private Sub mnum1KbytesROM_Click()
Form2.Smnu1KbytesROM
mnum2KbytesROM.Checked = False
mnum1KbytesROM.Checked = True
mnum4KbytesROM.Checked = False
End Sub

Private Sub mnum2_Click()
Form2.Smnu2
mnum1.Checked = False
mnum2.Checked = True
mnum3.Checked = False
mnum4.Checked = False
End Sub

Private Sub mnum2KbytesRAM_Click()
Form2.Smnu2KbytesRAM
mnum2KbytesRAM.Checked = True
mnum1KbytesRAM.Checked = False
mnum4KbytesRAM.Checked = False
End Sub

Private Sub mnum2KbytesROM_Click()
Form2.Smnu2KbytesROM
mnum2KbytesROM.Checked = True
mnum1KbytesROM.Checked = False
mnum4KbytesROM.Checked = False
End Sub

Private Sub mnum3_Click()
Form2.Smnu3
mnum1.Checked = False
mnum2.Checked = False
mnum3.Checked = True
mnum4.Checked = False
```

```
End Sub

Private Sub mnum4_Click()
Form2.Smnu4
mnum1.Checked = False
mnum2.Checked = False
mnum3.Checked = False
mnum4.Checked = True
End Sub

Private Sub mnum4KbytesRAM_Click()
Form2.Smnu4KbytesRAM
mnum2KbytesRAM.Checked = False
mnum1KbytesRAM.Checked = False
mnum4KbytesRAM.Checked = True
End Sub

Private Sub mnum4KbytesROM_Click()
Form2.Smnu4KbytesROM
mnum4KbytesROM.Checked = True
mnum1KbytesROM.Checked = False
mnum2KbytesROM.Checked = False
End Sub

Private Sub mnumAbrir_Click()
Form2.SmnuAbrir
mnumEjecutar.Enabled = True
mnumCodigo.Enabled = True
mnumCodigoDetallado.Enabled = True
mnumPorInstruccion.Enabled = True
mnumPorInstruccionSA.Enabled = True
mnumTodo.Enabled = True
mnumTodoSA.Enabled = True
Toolbar1.Buttons(3).Enabled = True
Toolbar1.Buttons(6).Enabled = True
Toolbar1.Buttons(7).Enabled = True
Toolbar1.Buttons(8).Enabled = True
Toolbar1.Buttons(9).Enabled = True
mnumAbrir.Enabled = False
Toolbar1.Buttons(1).Enabled = False
mnumCerrar.Enabled = True
mnumPorticoSerial.Enabled = False
End Sub

Private Sub mnumAcercadeMicroSolution_Click()
frmAcercaDe.Show
End Sub

Private Sub mnumAlgunaInstruccion_Click()
Form2.SmnuAlgunaInstruccion
End Sub

Private Sub mnumBuffer541a_Click()
frmBuffer244.Show
End Sub

Private Sub mnumCerrar_Click()
Unload frmDiagrama2
Unload Form2
Unload frm2KRam
Unload frm2KRom
Unload frmBuffer244
Unload frmCodigo
Unload frmCodigoDetallado
Unload frmDecodificador138
Unload frmDirec_Ram
Unload frmEdit_Ins
Unload frmLatch
Unload frmLatch373
Unload frmMemoriaROM
Unload frmMemorias
```

```
Unload frmMicro
Unload frmRamExterna
Unload frmSwitcheo_P1
Form2.Show
Form2.Hide
frmDiagrama2.Top = 0
frmDiagrama2.Left = 0
frmDiagrama2.Height = 7450
frmDiagrama2.Width = 11870
mnumAbrir.Enabled = True
Toolbar1.Buttons(1).Enabled = True
mnumPorticoSerial.Enabled = True
mnumCodigo.Enabled = False
mnumCodigoDetallado.Enabled = False
mnumEjecutar.Enabled = False
mnumCerrar.Enabled = False
Toolbar1.Buttons(3).Enabled = False
Toolbar1.Buttons(6).Enabled = False
Toolbar1.Buttons(7).Enabled = False
Toolbar1.Buttons(8).Enabled = False
Toolbar1.Buttons(9).Enabled = False
End Sub

Private Sub mnumCodigo_Click()
Form2.SmnuCodigo
End Sub

Private Sub mnumCodigoDetallado_Click()
frmCodigoDetallado.Show
End Sub

Private Sub mnumDecodificaciondeRAM_1_Click()
frmDirec_Ram.Show
End Sub

Private Sub mnumDecodificador138_Click()
frmDecodificador138.Show
End Sub

Private Sub mnumEditarInstruccion_Click()
frmEdit_Ins.Show
End Sub

Private Sub mnumFinalizar_Click()
Unload Form2
Form2.Show
Form2.Hide
mnumIniciar.Enabled = True
mnumAbrir.Enabled = True
Toolbar1.Buttons(1).Enabled = True
mnumCerrar.Enabled = False
mnumFinalizar.Enabled = False
mnumCodigo.Enabled = False
Toolbar1.Buttons(3).Enabled = False
mnumCodigoDetallado.Enabled = False
mnumDecodificador138.Enabled = True
mnumLatch377z.Enabled = True
mnumBuffer541a.Enabled = True
Toolbar1.Buttons(10).Enabled = True
mnum1KbytesROM.Enabled = True
mnum2KbytesROM.Enabled = True
mnum4KbytesROM.Enabled = True
mnum1KbytesRAM.Enabled = True
mnum2KbytesRAM.Enabled = True
mnum4KbytesRAM.Enabled = True
Form2.mnu1KbytesROM.Enabled = True
Form2.mnu2KbytesROM.Enabled = True
Form2.mnu4KbytesROM.Enabled = True
Form2.mnu1KbytesRAM.Enabled = True
Form2.mnu2KbytesRAM.Enabled = True
Form2.mnu4KbytesRAM.Enabled = True
```

```
mnumEdicion.Enabled = True
Toolbar1.Buttons(15).Enabled = True
Toolbar1.Buttons(16).Enabled = True
Toolbar1.Buttons(6).Enabled = False
Toolbar1.Buttons(7).Enabled = False
Unload frmDiagrama1
frmDiagrama2.Top = 0
frmDiagrama2.Left = 0
frmDiagrama2.Height = 7450
frmDiagrama2.Width = 11870
frmDiagrama2.Show
frmDiagrama1.Visible = False
Form2.mnuBuffer.Checked = True
Step = 0
ejec1 = -1
End Sub

Private Sub mnumIniciar_Click()
Unload Form2
Form2.Show
Form2.Hide
mnumIniciar.Enabled = False
mnumAbrir.Enabled = False
Toolbar1.Buttons(1).Enabled = False
mnumCerrar.Enabled = False
mnumFinalizar.Enabled = True
mnumCodigo.Enabled = True
Toolbar1.Buttons(3).Enabled = True
mnumCodigoDetallado.Enabled = True
mnumDecodificador138.Enabled = False
mnumLatch377z.Enabled = False
mnumBuffer541a.Enabled = False
Toolbar1.Buttons(10).Enabled = False
mnum1KbytesROM.Enabled = False
mnum2KbytesROM.Enabled = False
mnum4KbytesROM.Enabled = False
mnum1KbytesRAM.Enabled = False
mnum2KbytesRAM.Enabled = False
mnum4KbytesRAM.Enabled = False
Form2.mnu1KbytesROM.Enabled = False
Form2.mnu2KbytesROM.Enabled = False
Form2.mnu4KbytesROM.Enabled = False
Form2.mnu1KbytesRAM.Enabled = False
Form2.mnu2KbytesRAM.Enabled = False
Form2.mnu4KbytesRAM.Enabled = False
mnumEdicion.Enabled = False
Toolbar1.Buttons(15).Enabled = False
Toolbar1.Buttons(16).Enabled = False
Unload frmDiagrama2
frmDiagrama1.Top = 0
frmDiagrama1.Left = 0
frmDiagrama1.Height = 7450
frmDiagrama1.Width = 11870
frmDiagrama1.Show
frmDiagrama2.Visible = False
Form2.mnuBuffer.Checked = False
'Cargamos las instrucciones de "simulación" del Pórtico Serial
Form2.SmnuAbrir
Toolbar1.Buttons(6).Enabled = True
Toolbar1.Buttons(7).Enabled = True
End Sub

Private Sub mnumIntel8X51_Click()
Valret = Shell("C:\WINDOWS\WINHELP.EXE C:\8031\INTEL\M51fx.HLP", 1)
End Sub

Private Sub mnumInterruptores_P1_Click()
Form2.SmnuInterruptores_P1
End Sub

Private Sub mnumInterruptoresBuffer541_Click()
```

```
Form2.SmnuInterruptores_Buffer541
End Sub

Private Sub mnumLatch377a_Click()
frmLatch.Show
End Sub

Private Sub mnumLatch377s_Click()
Form2.SmnuLatch377s
End Sub

Private Sub mnumLatch377z_Click()
frmLatch373.Show
End Sub

Private Sub mnumMicrocontrolador_Click()
frmMicro.Show
End Sub

Private Sub mnumPorInstruccion_Click()
Form2.SmnuPorInstruccion
End Sub

Private Sub mnumPorInstruccionSA_Click()
frmDiagrama2.label1.Caption = saMatrizInstrucciones(ejec1 + 1)
frmDiagrama2.Label2.Caption = "#Bytes = " & sInstruccionesOperandos(ejec1 + 1, 8)
frmDiagrama2.Label3.Caption = "#C.Maq = " & sInstruccionesOperandos(ejec1 + 1, 9)
frmDiagrama2.label1.Visible = True
frmDiagrama2.Label2.Visible = True
frmDiagrama2.Label3.Visible = True
frmDiagrama2.Refresh
Form2.SmnuPorInstruccionSA
frmDiagrama2.label1.Visible = False
frmDiagrama2.Label2.Visible = False
frmDiagrama2.Label3.Visible = False
frmDiagrama2.Refresh
End Sub

Private Sub mnumRAM_Click()
Form2.SmnuVer_EsquematicoRam
End Sub

Private Sub mnumRAMInterna_Click()
frmMemorias.Show
End Sub

Private Sub mnumROM_Click()
Form2.SmnuVer_EsquematicoRom
End Sub

Private Sub mnumSalir_Click()
Unload Me
End
End Sub

Private Sub mnumTodo_Click()
Form2.SmnuTodo
End Sub

Private Sub mnumTodoSA_Click()
Form2.SmnuTodoSA
End Sub

Private Sub mnumVer_ContenidoRAM_Click()
frmRamExterna.Show
frmRamExterna.txtDE.SetFocus
End Sub

Private Sub mnumVer_ContenidoROM_Click()
frmMemoriaROM.Show
End Sub
```

```
Private Sub mnumVer_EsquematicoRAM_Click()
Form2.SmnuVer_EsquematicoRam
End Sub

Private Sub mnumVer_EsquemáticoROM_Click()
Form2.SmnuVer_EsquematicoRom
End Sub

Private Sub Toolbar1_ButtonClick(ByVal Button As Button)
If Button.Tag = "Abrir" Then
   mnumAbrir_Click
ElseIf Button.Tag = "PasoaPaso" Then
   mnumPorInstruccion_Click
ElseIf Button.Tag = "Codigo" Then
   mnumCodigo_Click
ElseIf Button.Tag = "Todo" Then
   mnumTodo_Click
ElseIf Button.Tag = "PasoaPasoSA" Then
   mnumPorInstruccionSA_Click
ElseIf Button.Tag = "TodoSA" Then
   mnumTodoSA_Click
ElseIf Button.Tag = "RamInterna" Then
   mnumRAMInterna_Click
ElseIf Button.Tag = "EscaladeTiempo" Then
   PopupMenu mnumEscaladeTiempo, vbPopupMenuLeftAlign
ElseIf Button.Tag = "RamExterna" Then
   mnumVer_ContenidoRAM_Click
ElseIf Button.Tag = "Rom" Then
   mnumVer_ContenidoROM_Click
ElseIf Button.Tag = "EditordeInstrucciones" Then
   mnumEditarInstruccion_Click
ElseIf Button.Tag = "DecodificacíondeRAM" Then
   mnumDecodificacíondeRAM_1_Click
End If
End Sub
```

MÓDULO1

```
'Declaración de variables
'Las siguientes variables se utilizan para el codigo
'de conversion de Decimal a Hexadecimal
Public iNumerosH(1) As Integer 'Tiene dos elementos (0) , (1)
Public sNumerosH(1) As String
Public sValorH As String   'Contiene el dato convertido a hexadecimal
Public iX As Long
Public iY As Integer, iZ As Integer
'Las siguientes variables se utilizan para el codigo
'de conversion de Decimal a Binario
Public iX1d_b As Byte
Public iX2d_b As Byte
Public iX3d_b As Byte
Public iMatrizB(7) As Byte 'Contiene los datos como bits, se encuentran con
                    'su bit memos significativo en (7)
                    'hasta el mas significativo en (0)
'Las siguientes variables se utilizan para el codigo
'de conversion de Binario a Decimal
Public iValorD As Integer 'Contine el valor en decimal
                   'resultante de la conversion
Public iMatrizB_D(7) As Byte   'Esta matiz contiene los datos
                     'equivalentes a 1,2,4,8,16,32,64,128
                     'de (7) a .... (0)
Public iValorB(7) As Byte  'Esta matriz contiene el dato en forma
                    'binaria a ser convertido a decimal
'Las sigientes variables se usan para el codigo
'de conversion Hexadecimal a decimal
Public sValorH_D As String  'Contiene el valor hexadecimal
Public iMValorH_D(1) As Byte   'Contiene el valor decimal del digito hexadecimal correspondiente
Public iValorH_D As Byte  'Guardará el dato decimal resultante
Public sValorCaracter(1) As String 'Contendra los datos que se vayan
```

```
                    'leyendo de sValorH_D
'Variables de la memoria RAM Interna
Public iMemoriaRAMint(255, 7) As Byte 'Contiene los datos de la memoria RAM como bits
                          'los datos del DPL y DPH tambien estaran aqui
Public iMemoriaRAMintD(255) As Byte  'Contiene los datos de la memoria RAM en forma decimal
                         'memos del DPH y DPL
Public iDPTRRamD As Long   'Guarda el valor del DPTR en forma decimal del total
                    'de los 16 bits del DPH y DPL
Public sMDPTRH(3) As String 'Guarda los 4 caracteres hexadecimales resultantes de la conversion
                     decimal a hexadecimal
Public iMDPTRH(3) As Integer 'Guarda los 4 numeros hexadecimales resultantes de la conversion
          decimal a hexadecirnal
Public i16Bits1 As Long, i16Bits2 As Long, i16Bits3 As Long
                    'Variables a utlizarce en la conversion del DPTR
                    'de decimal a binario
Public iBitDPTR(15) As Byte
Public sMemoriaRAMintH(255) As String  'Contiene los datos de la memoria RAM y DPTR en forma
                          'hexadecimal
                          'en la localidad sMemoriaRamintH(130) esta el valor del DPRT
                          'como hexadecimal completo
'Variables utilizadas para la conversión de datos cuando se trabaja
'con 16 bits
Public iValor16BitsD As Long   'Contiene el dato en forma decimal
   'Tambien se usa la variable iBitDPTR(15) ya declarada
   'sValorH_D
   'iX
   'sMDPTRH(3)
   'iMDPTRH(3)
Public sValor16BitsH As String  'Contiene el dato en forma hexadecimal
Public PC As Integer   'Funciona como contador del programa
Public SP As Integer   'Funciona como el Stack Pointer en el programa
                 'se lo inicia con 07H,por ello iniciará con 08H
                 'contendrá las direcciones como datos decimales
                 'su ubicación en la memoria Ram interna es 81H
Public SP1 As Integer  'Guarda el valor de la localidad de memoria ROM
                 'externa cuando se llama a una subrutina
Public ejec1 As Integer   'Se utilizan para saber la fila
Public ejec2 As Integer   'de la matriz de sInstruccionesOperandos
'Variables de la memoria RAM Externa y ROM Externa
Public iMemoriaRAMext(4095, 7) As Byte  'Contienen los datos de estas memorias externas
Public iMemoriaROMext(4095, 7) As Byte  'como bits
Public saMatrizInstrucciones(2000) As String 'Contiene todas las lineas de instrucciones leidas
'Memoria de instruccciones, opcode,operandos, bytes, #bytes, ciclos de máquina,
'direccionamiento en los porticos P2 y P0
Public sInstruccionesOperandos(2000, 25) As String 'Contine unicamente las lineas de instrucciones
                              'dividida en tipo de instruccion y operandos(5 columnas)
                              'Contiene también los bytes de cada instrucción (5 columnas)
Public sAddP0 As String   'Estas variables se usaran para el manejo de direcciones
Public sAddP2 As String   'y datos de instrucciones MOVX y MOVC
Public sDatP0 As String
Public sFileName1 As String
Public Frecuencia_Oscilador As Byte 'Usada para determinar
                         'la velocidad de simulación
                         'con animación en la variable
                         'tiem
Public iValor_In_P1 As Byte 'Es el valor que se switchea externamente en P1,
Public saValor_InB_P1(7) As Byte   'en formato decinal, binario y hexadecimal
Public sValor_InH_P1 As String
Public iValor_In_Buffer As Byte 'Es el valor que se switchea externamente en el Buffer,
Public saValor_InB_Buffer(7) As Byte   'en formato decinal, binario y hexadecimal
Public sValor_InH_Buffer As String
Public saValor_Latch377(7) As Byte  'Es el valor que queda en el Latch 377 como bits
Public iLongitud1 As Integer 'Para el número de carácteres de la línea leida
Public LineaLeida1 As String  'Para guardar los caracteres de toda la linea leida
Public LetrasLeidas As String  'Para guardar los caracteres leidos en un instante dado
Public iNum_Data As Byte 'Variable utilizada en Subrutina Num_Data
             'para guardar el valor leido #data
Public iRamBufferLatch As Byte  'Se usa para definir si se accesa a la memoria RAM
Public iRamBufferLatch1 As Byte  'al Buffer o al Latch dependiendo del valor de las direcciones
                         'segun los rangos
                         'RAM 8192 a 16383      1
```

```
                    'Buffer 16384 a 24575   2
                    'Latch 24576 a 32767    3
Public iBuffer_P1 As Boolean    'Para determinar si se hizo clic derecho en
                    'el Puerto P1 o en el Buffer
Public iRam As Byte 'Usadas en el formulario frmDirec_Ram
Public iBuffer As Byte  'para ubicar las direcciones en base
Public iLatch As Byte   'a los bits P2.7 P2.6 y P2.5
Public iLineasEntrada As Integer    'Para contar el número de
Public iEscrituraP1 As Boolean  'Nos permite con su valor de Verdadero o
                    'Falso impedir que las instrucciones PUSH P1,
                    'JB P1.x,rel, JNB P1.x,rel hagan falsas escrituras
                    'Leds del Puerto P1, en la Subrutina EscrituraP1()
Public iLecturaP1 As Byte   'Nos permite con su valor de Verdadero o
                    'Falso permitir que las instrucciones PUSH P1,
                    'JB P1.x,rel, JNB P1.x,rel hagan lecturas de los
                    'Leds del Puerto P1, en la Subrutina LecturaP1()
Public iMatriz As Integer
'Variable para manejar el 'numero de bytes que se seleccione para la memoria Ram
Public KbytesRam As Integer
Public iNum_Data16 As Long 'Se usa en la subrutina Num_Data16
                    'es un dato del tipo decimal
' Variables para controlar el Stack Pointer
Public iMatrizSP(30) As Integer 'Contenido Decimal
Public iFilasSP As Integer
'Variable de inicio para ejecución paso a paso del programa
Public Step As Integer
'Variables que se usaran para la instrucción MOVC
Public iRomc1 As Byte
Public KbytesRom As Integer
'Para manejo de los TIMERs como Contadores
Public ETimer0 As Boolean
Public ETimer1 As Boolean
Public STimer0 As Boolean
Public STimer1 As Boolean
'Para P1 en frmDiagrama1
Public iValor_In_P11 As Byte 'Es el valor que se switchea externamente en P1,
Public saValor_InB_P11(7) As Byte    'en formato decinal, binario y hexadecimal
Public sValor_InH_P11 As String
'Para Entrada Serial en frmDiagrama1
Public iValor_In_InS As Byte
Public saValor_InB_InS(7) As Byte
Public sValor_InH_InS As String

Sub Acc_Paridad()
'Esta subrrutina se utiliza para setear el bit P
'del registro PSW, que identifica el número de 1
'en el acumulador
par2 = 0
For par1 = 0 To 7
   If iMemoriaRAMint(224, par1) = 1 Then
      par2 = par2 + 1
   End If
Next par1
Select Case par2
   Case 1, 3, 5, 7
      iMemoriaRAMint(208, 7) = 1
   Case 0, 2, 4, 6, 8
      iMemoriaRAMint(208, 7) = 0
End Select
'Binario a Decimal
For n = 0 To 7
   iValorB(n) = iMemoriaRAMint(208, n)
Next n
ConversionB_D
iMemoriaRAMintD(208) = iValorD
'Decimal a Hexadecimal
iX = iMemoriaRAMintD(208)
ConversionD_H
sMemoriaRAMintH(208) = sValorH
End Sub
```

```
Sub ActualizacionDPTRB_DH()
'De binario a decimal
For adptr = 0 To 7
   iBitDPTR(adptr) = iMemoriaRAMint(131, adptr)
Next adptr
For adptr = 8 To 15
   iBitDPTR(adptr) = iMemoriaRAMint(130, adptr - 8)
Next adptr
iValor16BitsD = 0
ConversionDPTRB_D
iDPTRRamD = iValor16BitsD
'De decimal a hexadecimal
iX = iDPTRRamD
ConversionDPTRD_H
sMemoriaRAMintH(130) = sValor16BitsH
End Sub

Sub ActualizacionDPTRD_BH()
'De decimal a binaria
i16Bits1 = iDPTRRamD
ConversionDPTRD_B
For adptr1 = 0 To 7
   iMemoriaRAMint(131, adptr1) = iBitDPTR(adptr1)
Next adptr1
For adptr1 = 8 To 15
   iMemoriaRAMint(130, adptr1 - 8) = iBitDPTR(adptr1)
Next adptr1
'De decimal a hexadecimal
iX = iDPTRRamD
ConversionDPTRD_H
sMemoriaRAMintH(130) = sValor16BitsH
End Sub

Sub ActualizacionRAM()
'Las siguientes líneas permiten convertir los datos de la
'memoria RAM interna de binarios a decimales y hexadecimales
'De Binarios a Decimales
For m = 0 To 129
   For n = 0 To 7
      iValorB(n) = iMemoriaRAMint(m, n)
   Next n
   ConversionB_D
   iMemoriaRAMintD(m) = iValorD
Next m
For m = 132 To 255  'La division en dos ciclos For es necesaria para manejar
   For n = 0 To 7  'de manera separada los 16 bit del DPTR
      iValorB(n) = iMemoriaRAMint(m, n) 'tomar en cuenta lo dicho para todo
   Next n              'codigo que este involucrado con actualizacion de datos en la
      ConversionB_D   'memoria RAM
   iMemoriaRAMintD(m) = iValorD
Next m
'De Decimales a Hexadecimales
For mn = 0 To 129
   iX = iMemoriaRAMintD(mn)
   ConversionD_H
   sMemoriaRAMintH(mn) = sValorH
Next mn
For mn = 132 To 255
   iX = iMemoriaRAMintD(mn)
   ConversionD_H
   sMemoriaRAMintH(mn) = sValorH
Next mn
ActualizacionDPTRB_DH
'Controlamos la paridad del Acumulador
Acc_Paridad
End Sub

Sub ActualizacionRAMD_BH()
'Las siguientes líneas permiten convertir los datos de la
'memoria RAM interna de decimales a binarios y hexadecimales
'De Decimales a binarios
```

```
For md = 0 To 129
   iX1d_b = iMemoriaRAMintD(md)
   ConversionD_B
   For nd = 0 To 7
      iMemoriaRAMint(md, nd) = iMatrizB(nd)
   Next nd
Next md
For md = 132 To 255
   iX1d_b = iMemoriaRAMintD(md)
   ConversionD_B
   For nd = 0 To 7
      iMemoriaRAMint(md, nd) = iMatrizB(nd)
   Next nd
Next md
'De Decimales a Hexadecimales
For mnd = 0 To 129
   iX = iMemoriaRAMintD(mnd)
   ConversionD_H
   sMemoriaRAMintH(mnd) = sValorH
Next mnd
For mnd = 132 To 255
   iX = iMemoriaRAMintD(mnd)
   ConversionD_H
   sMemoriaRAMintH(mnd) = sValorH
Next mnd
ActualizacionDPTRD_BH
'Controlamos la paridad del Acumulador
Acc_Paridad
End Sub

Sub ConversionB_D()
iMatrizB_D(7) = 1
iMatrizB_D(6) = 2
iMatrizB_D(5) = 4
iMatrizB_D(4) = 8
iMatrizB_D(3) = 16
iMatrizB_D(2) = 32
iMatrizB_D(1) = 64
iMatrizB_D(0) = 128
iValorD = 0
For i = 0 To 7
   If iValorB(i) = 1 Then
      iValorD = iValorD + iMatrizB_D(i)
   End If
Next i
End Sub

Sub ConversionD_B()
For i = 0 To 7
   iMatrizB(i) = 0
Next i
If iX1d_b <> 0 Then
   i = 7
   While iX1d_b <> 1
      iX2d_b = iX1d_b \ 2
      iX3d_b = iX1d_b - 2 * iX2d_b
      iMatrizB(i) = iX3d_b
      i = i - 1
      iX1d_b = iX2d_b
   Wend
   iMatrizB(i) = 1
End If
End Sub

Sub ConversionD_H()
If iX >= 16 Then
   iY = iX \ 16
   iNumerosH(0) = iY
   iNumerosH(1) = iX - iY * 16
Else
   iNumerosH(0) = 0
```

```
    iNumerosH(1) = iX
  End If
  For i = 0 To 1
  Select Case iNumerosH(i)
    Case 15
      sNumerosH(i) = "F"
    Case 14
      sNumerosH(i) = "E"
    Case 13
      sNumerosH(i) = "D"
    Case 12
      sNumerosH(i) = "C"
    Case 11
      sNumerosH(i) = "B"
    Case 10
      sNumerosH(i) = "A"
    Case 0 To 9
      sNumerosH(i) = iNumerosH(i)
    End Select
  Next
  sValorH = sNumerosH(0) & sNumerosH(1)
  End Sub

  Sub ConversionDPTRB_D()
  iValor16BitsD = 0
  iMatrizB_D(7) = 1
  iMatrizB_D(6) = 2
  iMatrizB_D(5) = 4
  iMatrizB_D(4) = 8
  iMatrizB_D(3) = 16
  iMatrizB_D(2) = 32
  iMatrizB_D(1) = 64
  iMatrizB_D(0) = 128
  For dp = 8 To 15
    If iBitDPTR(dp) = 1 Then
      iValor16BitsD = iValor16BitsD + iMatrizB_D(dp - 8)
    End If
  Next dp
  For dp = 0 To 7
    If iBitDPTR(dp) = 1 Then
      iValor16BitsD = iValor16BitsD + (iMatrizB_D(dp) - 1) * 256 + 256
    End If
  Next dp
  End Sub

  Sub ConversionDPTRD_B()
  For xdptr = 0 To 15
    iBitDPTR(xdptr) = 0
  Next xdptr
  If i16Bits1 <> 0 Then
    xdptr2 = 15
    While i16Bits1 <> 1
      i16Bits2 = i16Bits1 \ 2
      i16Bits3 = i16Bits1 - i16Bits2 - i16Bits2
      iBitDPTR(xdptr2) = i16Bits3
      xdptr2 = xdptr2 - 1
      i16Bits1 = i16Bits2
    Wend
    iBitDPTR(xdptr2) = 1
  End If
  End Sub

  Sub ConversionDPTRD_H()
  If iX >= 16 Then
    If iX >= 256 Then
      If iX >= 4096 Then
        iY = iX \ 16
        iMDPTRH(3) = iX - iY * 8 - iY * 8
        iX = iY
        iY = iX \ 16
        iMDPTRH(2) = iX - iY * 16
```

```
            iX = iY
            iY = iX \ 16
            iMDPTRH(0) = iY
            iMDPTRH(1) = iX - iY * 16
        Else
            iMDPTRH(0) = 0
            iY = iX \ 16
            iMDPTRH(3) = iX - iY * 16
            iX = iY
            iY = iX \ 16
            iMDPTRH(1) = iY
            iMDPTRH(2) = iX - iY * 16
        End If
    Else
        iMDPTRH(0) = 0
        iMDPTRH(1) = 0
        iY = iX \ 16
        iMDPTRH(2) = iY
        iMDPTRH(3) = iX - iY * 16
    End If
Else
    iMDPTRH(0) = 0
    iMDPTRH(1) = 0
    iMDPTRH(2) = 0
    iMDPTRH(3) = iX
End If
For i = 0 To 3
Select Case iMDPTRH(i)
    Case 15
        sMDPTRH(i) = "F"
    Case 14
        sMDPTRH(i) = "E"
    Case 13
        sMDPTRH(i) = "D"
    Case 12
        sMDPTRH(i) = "C"
    Case 11
        sMDPTRH(i) = "B"
    Case 10
        sMDPTRH(i) = "A"
    Case 0 To 9
        sMDPTRH(i) = iMDPTRH(i)
    End Select
Next
sValor16BitsH = sMDPTRH(0) & sMDPTRH(1) & sMDPTRH(2) & sMDPTRH(3)
End Sub

Sub ConversionDPTRH_D()
For idh_d = 0 To 3
sMDPTRH(idh_d) = Mid(sValorH_D, idh_d + 1, 1)
sMDPTRH(idh_d) = UCase(sMDPTRH(idh_d))
    Select Case sMDPTRH(idh_d)
        Case "F"
            iMDPTRH(idh_d) = 15
        Case "E"
            iMDPTRH(idh_d) = 14
        Case "D"
            iMDPTRH(idh_d) = 13
        Case "C"
            iMDPTRH(idh_d) = 12
        Case "B"
            iMDPTRH(idh_d) = 11
        Case "A"
            iMDPTRH(idh_d) = 10
        Case Else
            iMDPTRH(idh_d) = Val(sMDPTRH(idh_d))
    End Select
Next idh_d
iValor16BitsD = ((iMDPTRH(0) * 16 + iMDPTRH(1)) * 16 + iMDPTRH(2)) * 8
iValor16BitsD = iValor16BitsD + ((iMDPTRH(0) * 16 + iMDPTRH(1)) * 16 + iMDPTRH(2)) * 8 + iMDPTRH(3)
End Sub
```

```
Sub ConversionH_D()
iValorH_D = 0
iMValorH_D(0) = 0
iMValorH_D(1) = 0
For ih_d = 0 To 1
sValorCaracter(ih_d) = Mid(sValorH_D, ih_d + 1, 1)
sValorCaracter(ih_d) = UCase(sValorCaracter(ih_d))
   Select Case sValorCaracter(ih_d)
      Case "F"
         iMValorH_D(ih_d) = 15
      Case "E"
         iMValorH_D(ih_d) = 14
      Case "D"
         iMValorH_D(ih_d) = 13
      Case "C"
         iMValorH_D(ih_d) = 12
      Case "B"
         iMValorH_D(ih_d) = 11
      Case "A"
         iMValorH_D(ih_d) = 10
      Case Else
         iMValorH_D(ih_d) = Val(sValorCaracter(ih_d))
   End Select
Next ih_d
iValorH_D = iMValorH_D(0) * 16 + iMValorH_D(1)
End Sub
```