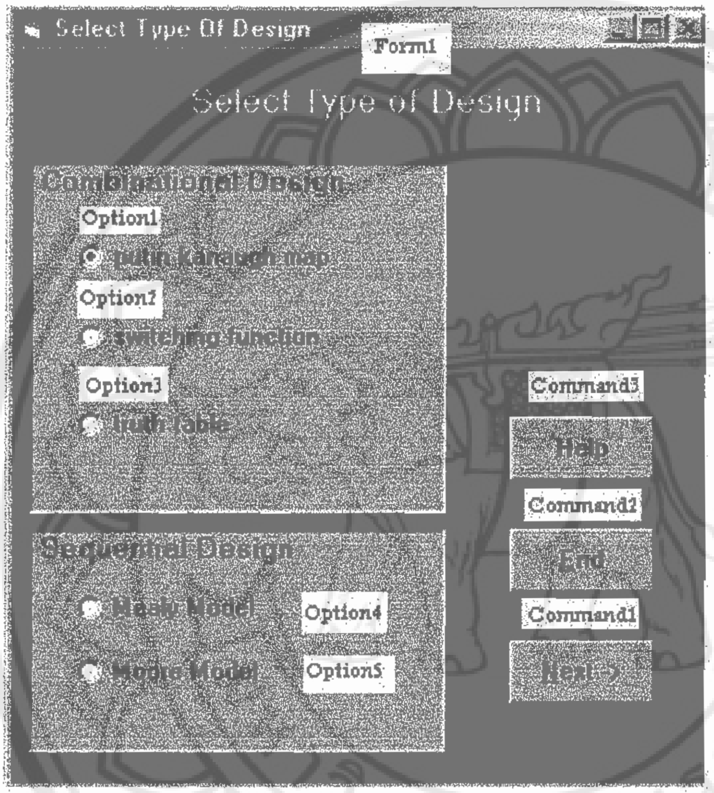
ภาคผนวก

ซ้อสโค๊ด(Source Code)โปรแกรมการออกแบบวงจรดิจิตอลเป็นดังนี้

Form1(ส่วนการเลือกชนิดของการออกแบบ)

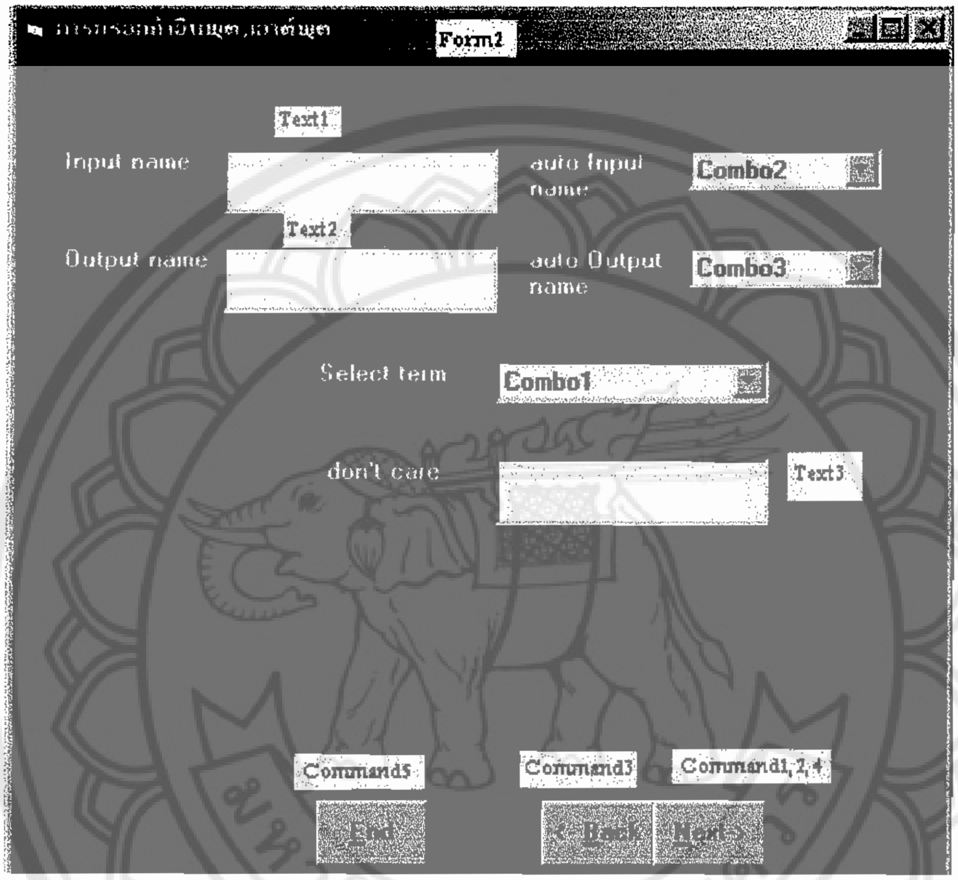


```
Public a As Integer
Private Sub Command1_Click ()
If (Option1.value = True) Or (Option2.value = True) Or (Option3.value = True) Then
Load Form2
Form2.Show
If Option1.value = True Then
Form2.Command1.Visible = True
End If
If Option2.value = True Then
Form2.Command2.Visible = True
```

```
End If

If Option3.value = True Then
Form2.Command4.Visible = True
Form2.Text3.Visible = True
Form2.Label6.Visible = True
End If
Else
Load Form7
Form7.Show
End If
Form1.Visible = False
End Sub

Private Sub Command2_Click()
End
End Sub

Private Sub Command3_Click()
Load Help
Help.Show
End Sub
```

Form2(ส่วนกรอกค่าอินพุต,เอาท์พุต)

```
Public numout As Integer
Public num As Integer
Public a As Integer
Dim ftextok As Integer
Dim stextok As Integer
Dim combook As Integer
Dim un() As String
Dim unv() As String
Dim sameok As Integer
```



```
Public Function cuts(st As String) As String()
Dim lon As Integer
Dim i As Integer
Dim ii As Integer
Dim numm As Integer
Dim puk As String
Dim va As String
Dim kk As String
Dim repeat As Integer
Dim unk() As String
ReDim unk(5)
```

```
repeat = 0
ftextok = 0
numm = 0
kk = ""
lon = Len(st)
If lon = 0 Then
MsgBox ("คุณต้องใส่ตัวแปร")
Exit Function
Else
For i = 1 To lon
puk = Left$(st, i)
va = Right$(puk, 1)
If va = "," Then
If kk <> "" Then
unk(numm) = kk
numm = numm + 1
kk = ""
Else
MsgBox ("คุณไม่ใส่ตัวแปรหรือผิดหลักโปรแกรม")
Exit Function
End If
Else
If va <> " " Then
If Not ((va >= "0") And (va <= "9")) Then
kk = kk & va
Else
MsgBox ("you should not define input,output is number")
Exit Function
End If
End If
End If
Next I
End If
If kk <> "" Then
unk(numm) = kk
numm = numm + 1
End If

If st Like Text1.Text Then
If numm = 1 Then
MsgBox ("you have input 2 to 5")
Exit Function
End If
If numm >= 6 Then
MsgBox ("you have input 2 to 5")
Exit Function
End If
Else
If numm > 4 Then
MsgBox ("you have output 1 to 4")
Exit Function
End If
End If
For i = 0 To numm - 2
For ii = i + 1 To numm - 1
If unk(i) = unk(ii) Then
repeat = 1
End If
Next ii
Next i
```

```
If repeat = 1 Then
MsgBox ("you can not put same input or output")
Exit Function
End If

If st Like Text1.Text Then
num = numm
nu = numm
Else
numout = numm
no = numm
End If
ftextok = 1
cuts = unk
End Function

Sub checkcombo()
combook = 0

If Combo1.Text <> "" Then
 combook = 1
Else
MsgBox ("you must choose term")
End If
End Sub

Sub checksame()
Dim i As Integer
Dim ii As Integer
sameok = 0
```

```
For i = 0 To num - 1
For ii = 0 To numout - 1

If unv(i) Like un(ii) Then
MsgBox ("inputname and outputname has same")
Exit Sub
End If

Next ii
Next i
sameok = 1
End Sub

Private Sub Combo2_Click()
Select Case Combo2.Text
Case "2"
Text1.Text = "A,B"
Case "3"
Text1.Text = "A,B,C"
Case "4"
Text1.Text = "A,B,C,D"
Case "5"
Text1.Text = "A,B,C,D,E"
End Select
End Sub

Private Sub Combo3_Click()
Select Case Combo3.Text
Case "1"
Text2.Text = "Z"
```

```
Case "2"
Text2.Text = "Y,Z"
Case "3"
Text2.Text = "X,Y,Z"
Case "4"
Text2.Text = "W,X,Y,Z"
End Select
End Sub

Private Sub Command1_Click()
num = 0
numout = 0

If (Text2.Text <> "") Or (Text1.Text <> "") Then
If Text1.Text Like Text2.Text Then
MsgBox ("inputname and outputname has same")
Exit Sub
End If
End If
unv = cuts(Text1.Text)

If ftextok = 1 Then
checkcombo
End If

If (ftextok = 1) And (combook = 1) Then
un = cuts(Text2.Text)
If ftextok = 1 Then
checkcombo
End If
```

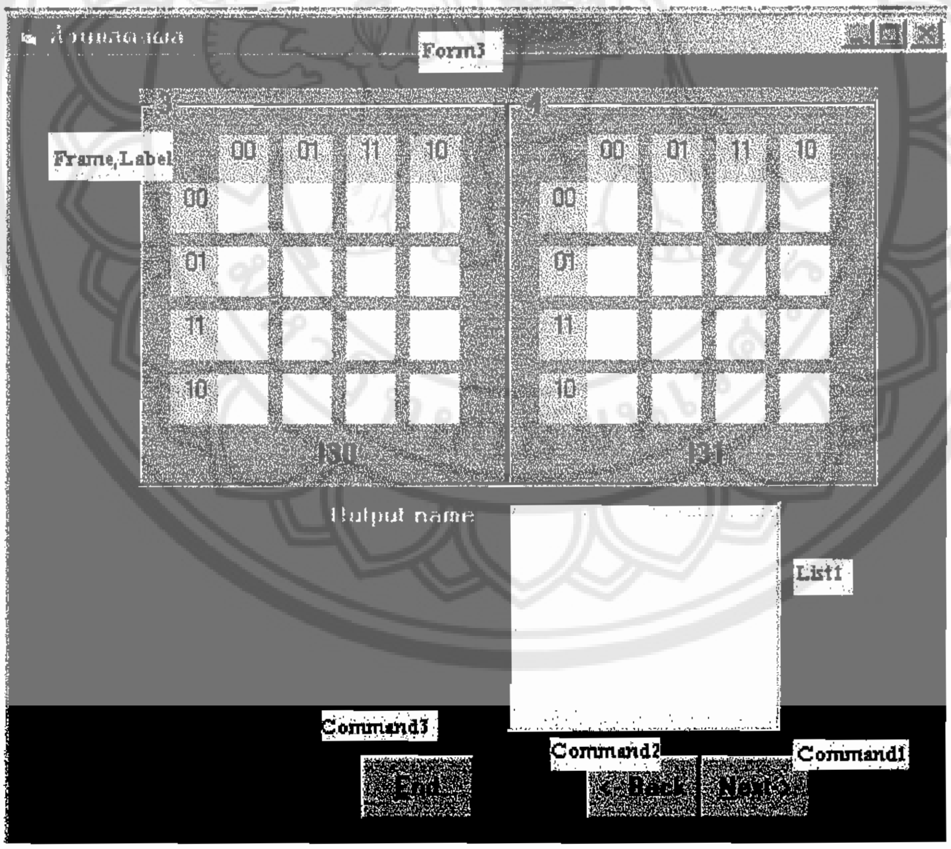
```
If (num > 0) And (numout > 0) Then
checksame
End If
If sameok = 1 Then
If (ftextok = 1) And (combook = 1) Then
Load Form3
Form3.Show
Form2.Visible = False
End If
End If
End If
End Sub

Private Sub Command2_Click()
Dim ss As String
Dim i As Integer
num = 0
numout = 0
If (Text2.Text <> "") Or (Text1.Text <> "") Then
If Text1.Text Like Text2.Text Then
MsgBox ("inputname and outputname has same")
Exit Sub
End If
End If

unv = cuts(Text1.Text)

If ftextok = 1 Then
checkcombo
End If
```

```
If (ftextok = 1) And (combook = 1) Then
un = cuts(Text2.Text)

If ftextok = 1 Then
checkcombo
End If
If (num > 0) And (numout > 0) Then
checksame
End If
If sameok = 1 Then
If (ftextok = 1) And (combook = 1) Then
 Load Form4
Form4.Show
Form2.Visible = False

For i = 0 To numout - 1
Form4.Label1(i).Visible = True
Form4.Label2(i).Visible = True
Form4.Text1(i).Visible = True
Form4.Text2(i).Visible = True
Form4.Label1(i).Caption = un(i)
Next i
For i = 0 To num - 1
ss = ss & unv(i)
If i <> num - 1 Then
ss = ss & ","
End If
Next i
For i = 0 To 3
```

```
Form4.Label1(i).Caption = Form4.Label1(i).Caption & "(" & ss & ")"
Next i
End If
End If
End If
End Sub

Private Sub Command3_Click()
Unload Form1
Load Form1
Form1.Show
Unload Form2
End Sub

Private Sub Command4_Click()
Dim i As Integer
Dim dc As Integer
Dim dcar() As Integer
Dim ii As Integer
Dim aa As String
Dim bb As String
num = 0
numout = 0

If (Text2.Text <> "") Or (Text1.Text <> "") Then
If Text1.Text Like Text2.Text Then
MsgBox ("inputname and outputname has same")
Exit Sub
End If
End If
```

```
unv = cuts(Text1.Text)

If ftextok = 1 Then
checkcombo
End If

If (ftextok = 1) And (combook = 1) Then
un = cuts(Text2.Text)
If ftextok = 1 Then
checkcombo
End If

If (num > 0) And (numout > 0) Then
Checksame
End If

If sameok = 1 Then
If (ftextok = 1) And (combook = 1) Then
dcar = Form4.cutnumd(Form2.Text3.Text)
dc = Form4.dc

If Form4.pa = 0 Then
Exit Sub
Else
Load Form5
Form5.Show
Form2.Visible = False
End If
End If
End If
End If
End Sub
Private Sub Command5_Click()
End
End Sub

Private Sub Form_Load()
num = 0
numout = 0
End Sub
```

Form3 (ส่วนแสดงผล)



```
Public logic As String
Dim firstx As Integer
Dim firsty As Integer
Dim xy As Integer
Dim frameindex As Integer
Dim arrays(12, 35) As Integer
Dim a(12) As Integer
Public numout As Integer
Dim output() As String
Dim dc(12) As Integer
Dim dcare(12, 35) As Integer
Dim unk() As String
Public num As Integer
Public tnumf As Integer
Public tnuml As Integer
Dim listout As Integer
Public aa As Integer
Public dcc As Integer

Sub locate()
Dim i As Integer
dc(listout) = -1
a(listout) = -1
For i = tnumf To tnuml
If (Label1(i).Caption = logic) Then
a(listout) = a(listout) + 1
arrays(listout, a(listout)) = i - tnumf
End If
If Label1(i).Caption = "d" Then
dc(listout) = dc(listout) + 1
dcare(listout, dc(listout)) = i
End If
Next I
```

```
End Sub
Sub appear()
Dim i As Integer

Select Case num
Case 2
Frame1(0).Visible = True
Frame1(0).Caption = unk(1) & " \ " & unk(0)
tnumf = 0
tnuml = 3
Case 3
Frame1(1).Visible = True
Frame1(1).Caption = unk(1) & " " & unk(2) & " \ " & unk(0)
tnumf = 4
tnuml = 11
Case 4
Frame1(2).Visible = True
Frame1(2).Caption = unk(2) & " " & unk(3) & " \ " & unk(0) & " " & unk(1)
tnumf = 12
tnuml = 27
Case 5
Frame1(2).Visible = True
Frame1(3).Visible = True
Frame1(2).Caption = unk(3) & " " & unk(4) & " \ " & unk(1) & " " & unk(2)
Frame1(3).Caption = unk(3) & " " & unk(4) & " \ " & unk(1) & " " & unk(2)
Label3(0).Visible = True
Label3(1).Visible = True
Label3(0).Caption = unk(0) & " = 0"
Label3(1).Caption = unk(0) & " = 1"
tnumf = 12
```

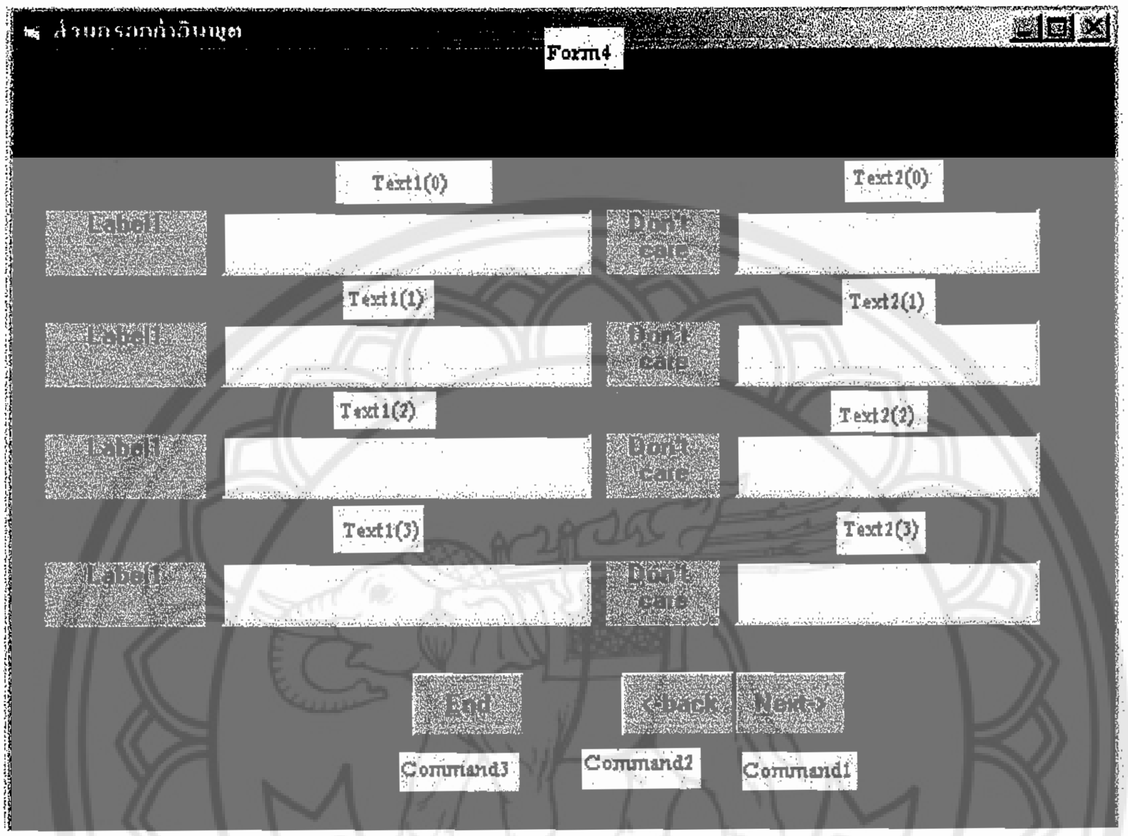
```
tnuml = 43
End Select
For i = 0 To numout - 1
List1.AddItem output(i)
Next i
End Sub

Sub showkn()
Dim i As Integer
For i = tnumf To tnuml
Label1(i).Caption = ""
Next i
For i = 0 To a(listout)
Label1(arrays(listout, i) + tnumf).Caption = logic
Next i
For i = 0 To dc(listout)
Label1(dcare(listout, i)).Caption = "d"
Next i
End Sub

Public Function continlogic(out As Integer) As Integer()
Dim i As Integer
Dim arr() As Integer
ReDim arr(33)
aa = a(out)
For i = 0 To a(out)
arr(i) = arrays(out, i)
Next i
continlogic = arr
End Function
```

```
Public Function contindc(out As Integer) As Integer()
Dim i As Integer
Dim dcar() As Integer
ReDim dcar(32)
dcc = dc(out)
For i = 0 To dc(out)
dcar(i) = dcare(out, i)
Next i
contindc = dcar
End Function

Private Sub Command1_Click()
locate
Load Form6
Form6.Show
Form3.Visible = False
End Sub

Private Sub Command2_Click()
Unload Form2
Load Form2
Form2.Show
Form2.Command1.Visible = True
Unload Form3
End Sub

Private Sub Command3_Click()
End
End Sub
```

```
Private Sub Form_DragDrop(Source As Control, X As Single, Y As Single)
Frame1(frameindex).Top = Y - firsty
Frame1(frameindex).Left = X - firstx
xy = 0
End Sub

Private Sub Form_Load()
Dim i As Integer
If Form2.Combo1.Text = "Minterm" Then
logic = "1"
Else
logic = "0"
End If

unk = Form2.cuts(Form2.Text1)
num = Form2.num
output = Form2.cuts(Form2.Text2)
numout = Form2.numout
appear
List1.ListIndex = 0
listout = 0

For i = 0 To numout - 1
a(i) = -1
dc(i) = -1
Next i
End Sub

Private Sub Form_Unload(Cancel As Integer)
Dim i As Integer
```

```
For i = 0 To numout - 1
List1.RemoveItem numout - 1 - i
Next i
End Sub
Private Sub Frame1_DragOver(Index As Integer, Source As Control, X As Single, Y As Single,
State As Integer)

If xy = 0 Then
frameindex = Index
firstx = X
firsty = Y
xy = 1
End If
End Sub

Private Sub Label1_Click(Index As Integer)
Select Case Label1(Index).Caption
Case ""
Label1(Index).Caption = logic
Case logic
Label1(Index).Caption = "d"
Case "d"
Label1(Index).Caption = ""
End Select
End Sub
Private Sub List1_Click()
locate
listout = List1.ListIndex
showkn
End Sub
```

Form4 (ส่วนกรอกค่าอินพุต)



```
Dim pass As Integer
Public pa As Integer
Public a As Integer
Public dc As Integer
Dim wo As Integer
Public Function cutnuml(st As String) As Integer()
Dim lon As Integer
Dim i As Integer
Dim puk As String
Dim va As String
Dim kk As String
Dim c As Integer
Dim repeat As Integer
Dim ii As Integer
Dim cnum As String
Dim vaa As Integer
Dim arrays() As Integer
ReDim arrays(32)
a = -1
kk = ""
lon = Len(st)
c = (2 ^ Form2.num) - 1
cnum = str$(c)

If lon = 0 Then
MsgBox ("you must put value")
Exit Function
Else
For i = 1 To lon
puk = Left$(st, i)
va = Right$(puk, 1)
If va = "," Then
If kk <> "" Then
vaa = Val(kk)
If (vaa >= 0) And (vaa <= c) Then
a = a + 1
arrays(a) = vaa
kk = ""
Else
MsgBox ("value must have 0 to " & cnum)
Exit Function
End If
Else
```

```
MsgBox ("คุณไม่ใส่ตัวแปรหรือผิดหลักโปรแกรม")
Exit Function
End If
Else
If (va >= "0") And (va <= "9") Then
kk = kk & va
Else
MsgBox ("you must put number")
Exit Function
End If
End If
Next i
End If

If kk <> "" Then
vaa = Val(kk)
If (vaa >= 0) And (vaa <= c) Then
a = a + 1
arrays(a) = vaa
kk = ""
Else
MsgBox ("value must have 0 to " & cnum)
Exit Function
End If
End If
For i = 0 To a - 1
For ii = i + 1 To a
If arrays(i) = arrays(ii) Then
repeat = 1
End If
```

```
Next ii

Next i

If repeat = 1 Then

MsgBox ("you can not put same value")

Exit Function

End If


If a >= 32 Then

MsgBox ("you have value not over 32")

Exit Function

End If

pass = pass + I

cutnuml = arrays

End Function


Public Function cutnumd(st As String) As Integer()

Dim lon As Integer

Dim i As Integer

Dim puk As String

Dim va As String

Dim kk As String

Dim c As Integer

Dim repeat As Integer

Dim ii As Integer

Dim cnum As String

Dim vaa As Integer

Dim arrays() As Integer

ReDim arrays(32)

dc = -1

kk = ""
```

```
pa = 0
Ion = Len(st)
c = (2 ^ Form2.num) - 1
cnum = str$(c)

If lon <> 0 Then
For i = 1 To lon
puk = Left$(st, i)
va = Right$(puk, 1)
If va = "," Then
If kk <> "" Then
vaa = Val(kk)
If (vaa >= 0) And (vaa <= c) Then
dc = dc + 1
arrays(dc) = vaa
kk = ""
Else
MsgBox ("value must have 0 to " & cnum)
Exit Function
End If
Else
MsgBox ("คุณไม่ใส่ตัวแปรหรือผิดหลักโปรแกรม")
Exit Function
End If
Else
If (va >= "0") And (va <= "9") Then
kk = kk & va
Else
MsgBox ("you must put number")
Exit Function
```

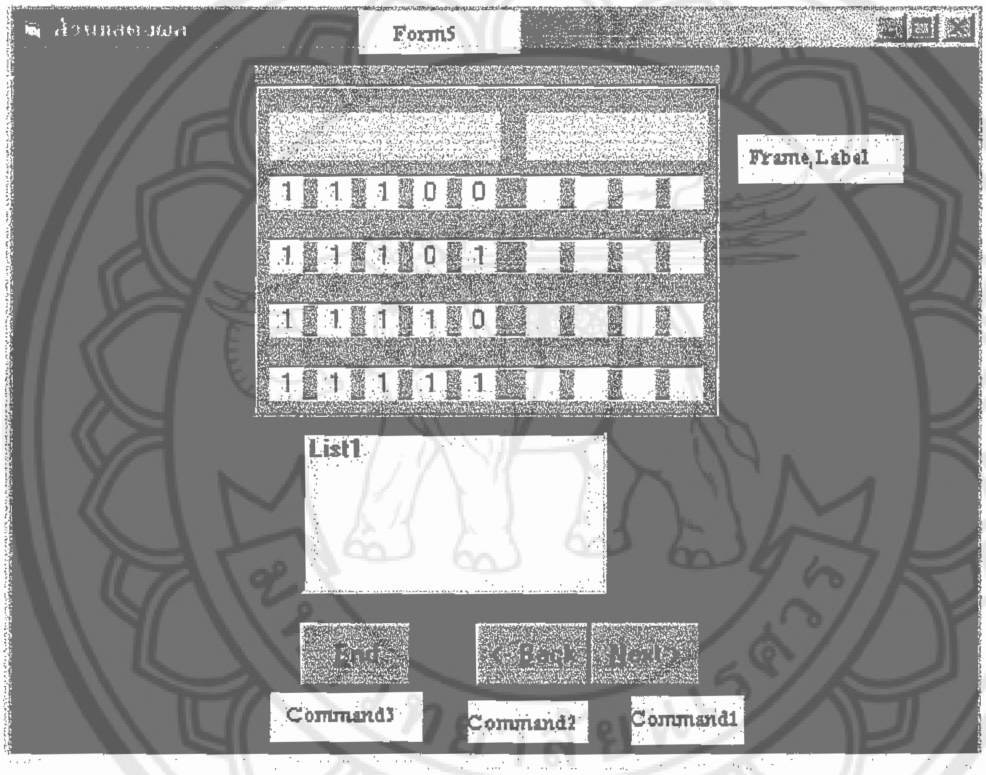
```
End If
End If
Next i
End If
If kk <> "" Then
vaa = Val(kk)
If (vaa >= 0) And (vaa <= c) Then
dc = dc + 1
arrays(dc) = vaa
kk = ""
Else
MsgBox ("value must have 0 to " & cnum)
Exit Function
End If
End If
For i = 0 To dc - 1
For ii = i + 1 To dc

If arrays(i) = arrays(ii) Then
repeat = 1
End If
Next ii
Next i

If repeat = 1 Then
MsgBox ("you can not put same value")
Exit Function
End If

If dc >= 32 Then
```

```
MsgBox ("you have value not over 32")
Exit Function
End If

pass = pass + 1
pa = 1
cutnumd = arrays
End Function

Private Sub Command1_Click()
Dim arr() As Integer
Dim dcar() As Integer
Dim i As Integer
Dim ii As Integer
Dim X As Integer
Dim Y As Integer
Dim same As Integer

pass = 0
same = 0
For i = 0 To (Form2.numout - 1)
wo = pass
arr = cutnuml(Text1(i).Text)
If wo <> pass Then
wo = pass
dcar = cutnumd(Text2(i).Text)
If wo = pass Then
Exit For
End If
Else
```

```
Exit For
End If

For X = 0 To a
For Y = 0 To dc
If arr(X) = dcar(Y) Then
same = 1
End If
Next Y
Next X
If same = 1 Then
MsgBox ("logic and don't care is same value")
Exit For
End If
Next i

If (pass = (2 * Form2.numout)) And (same = 0) Then
Load Form6
Form6.Show
Form4.Visible = False
End If

End Sub

Private Sub Command2_Click()
Unload Form2
Load Form2
Form2.Show
Form2.Command2.Visible = True
Unload Form4
```

```
End Sub
Private Sub Command3_Click()
End
End Sub
```

Form5 (ส่วนแสดงผล)



```
1 1 1 0 0
1 1 1 0 1
1 1 1 1 0
1 1 1 1 1
```

```
Dim dc As Integer
Dim dcar() As Integer
Dim num As Integer
Dim numout As Integer
Dim unk() As String
Dim output() As String
Dim oldlist As Integer

Private Sub Command1_Click()
```

```
Load Form6
Form6.Show
Form5.Visible = False
End Sub

Private Sub Command2_Click()
Unload Form2
Load Form2
Form2.Show
Form2.Command4.Visible = True
Form2.Text3.Visible = True
Form2.Label6.Visible = True
Unload Form5
End Sub

Private Sub Command3_Click()
End
End Sub

Private Sub Form_Load()
Dim i As Integer
Dim v As Integer
Dim st As String
Dim namein As String
Dim nameout As String
Dim ii As Integer
Dim log As String
Dim X As Integer
Dim oo As Integer
```

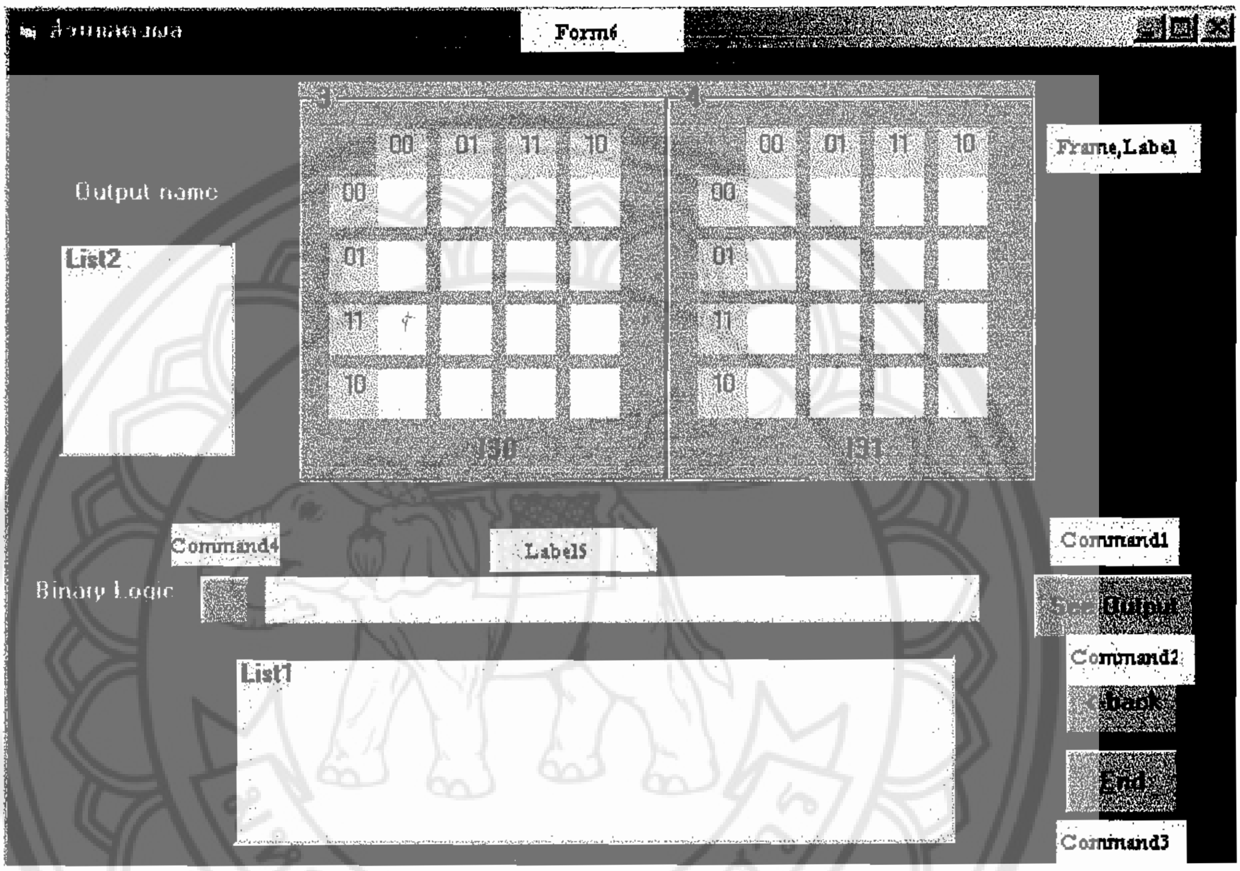
```
v = 0
unk = Form2.cuts(Form2.Text1.Text)
num = Form2.num
output = Form2.cuts(Form2.Text2.Text)
numout = Form2.numout

dcar = Form4.cutnumd(Form2.Text3.Text)
dc = Form4.dc
For i = 0 To num - 1
If i = num - 1 Then
namein = namein & unk(i)
Else
namein = namein & unk(i) & "  "
End If
Next i

For i = 0 To numout - 1
If i = numout - 1 Then
nameout = nameout & output(i)
Else
nameout = nameout & output(i) & "  "
End If
Next i
For i = 0 To (2 ^ (num - 2)) - 1
st = str$(v) & " - " & str$(v + 3)
List1.AddItem st
v = v + 4
Frame2(i).Caption = st
Label3(i).Caption = namein
Label4(i).Caption = nameout
```

```
Next i
For i = 0 To dc
Frame1(dcar(i)).Enabled = False
Next i
For i = 0 To num - 1
v = i
For ii = 0 To (2 ^ num) - 1
Label1(v).Visible = True
v = v + 5
Next ii
Next i

If Form2.Combo1.Text = "Minterm" Then
log = "0"
Else
log = "1"
End If
For i = 0 To numout - 1
v = i
For ii = 0 To (2 ^ num) - 1
Label2(v).Visible = True
oo = 0
For X = 0 To dc
If ii = dcar(X) Then
oo = 1
End If
Next X
If oo = 0 Then
Label2(v).Caption = log
End If
```

```
v = v + 4
Next ii
Next i
List1.ListIndex = 0
oldlist = 0
End Sub

Private Sub Label2_Click(Index As Integer)
Select Case Label2(Index).Caption
Case "0"
Label2(Index).Caption = "1"
Case "1"
Label2(Index).Caption = "0"
End Select
End Sub

Private Sub List1_Click()
Frame2(oldlist).Visible = False
Frame2(List1.ListIndex).Visible = True
oldlist = List1.ListIndex
End Sub
```

Form6 (ส่วนการแสดงผล)

```
Dim a(12) As Integer
Dim hj(I2) As Integer
Dim hjj(I2, 32) As Integer
Dim n(12) As Integer
Dim arrays(12, 32) As Integer
Dim strnum(12, 10, 100, 5) As Integer
Dim cycle(12, 100, 100) As Integer
Dim cy(12, 100) As Integer
Dim messp(12, 50) As String
```



```
Dim logic As String
Dim firstx As Integer
Dim firsty As Integer
Dim xy As Integer
Dim State() As String
Dim sta As Integer
Dim frameindex As Integer
Dim oldlist As Integer
Dim outff(5) As String
Dim outf As Integer
Dim inff(10) As String
Dim inf As Integer

Function checkdo(use() As Integer, u As Integer) As Integer
Dim z As Integer
Dim zz As Integer
Dim zi As Integer
Dim bb As Integer
Dim yes As Integer
Dim out As Integer
Dim str(30, 5) As Integer
Dim cyc(30, 35) As Integer
Dim cyy(35) As Integer
Dim h(32) As Integer

yes = 0
out = 0
For z = 0 To u
For zz = 0 To cy(lo, use(z))
For zi = 0 To hj(lo)
If cycle(lo, use(z), zz) = (hjj(lo, zi) + tnumf) Then
h(zi) = 1
End If
Next zi
Next zz
Next z
bb = -1
For z = 0 To hj(lo)
If h(z) = 1 Then
bb = bb + 1
End If
```

```
Next z
If bb = hj(lo) Then
yes = 1
out = 1
End If
If yes = 1 Then
For z = 0 To u
For zz = 0 To num - 1
str(z, zz) = strnum(lo, n(lo), use(z), zz)
Next zz
For zz = 0 To cy(lo, use(z))
cyc(z, zz) = cycle(lo, use(z), zz)
Next zz
cyy(z) = cy(lo, use(z))
Next z
For z = 0 To u
For zz = 0 To num - 1
strnum(lo, n(lo), z, zz) = str(z, zz)
Next zz
For zz = 0 To cyy(z)
cycle(lo, z, zz) = cyc(z, zz)
Next zz
cy(lo, z) = cyy(z)
Next z
k(lo) = u
End If
checkdo = out
End Function

Function updatecycle(u As Integer, use() As Integer)
```

```
Dim z As Integer
Dim zz As Integer
Dim cyc(30, 35) As Integer
Dim cyy(35) As Integer
Dim str(30, 5) As Integer

If u <> k(lo) Then
If u <> -1 Then
For z = 0 To u
For zz = 0 To num - 1
str(z, zz) = strnum(lo, n(lo), use(z), zz)
Next zz
For zz = 0 To cy(lo, use(z))
cyc(z, zz) = cycle(lo, use(z), zz)
Next zz
cyy(z) = cy(lo, use(z))
Next z
For z = 0 To u
For zz = 0 To num - 1
strnum(lo, n(lo), z, zz) = str(z, zz)
Next zz
For zz = 0 To cyy(z)
cycle(lo, z, zz) = cyc(z, zz)
Next zz
cy(lo, z) = cyy(z)
Next z
k(lo) = u
Else
a(lo) = -1
End If
End If
End Function

Sub checkdcare()
Dim i As Integer
Dim ii As Integer
Dim X As Integer
Dim numdon As Integer
Dim u As Integer
Dim use(20) As Integer
Dim z As Integer
Dim s As Integer
Dim oneuse(20) As Integer
Dim Noneuse As Integer
Dim has As Integer
Dim nub As Integer
Dim nubc As Integer
Dim cyc(35) As Integer
Dim onlyone(30, 35) As Integer
Dim Nouse(30) As Integer
Dim nu As Integer

u = -1
For i = 0 To k(lo)
numdon = -1
For ii = 0 To cy(lo, i)
For X = 0 To dc(lo)
If cycle(lo, i, ii) = dcare(lo, X) Then
numdon = numdon + 1
End If
```

```
Next X
Next ii
If numdon <> cy(lo, i) Then
u = u + 1
use(u) = i
End If
Next i
Call updatecycle(u, use())


For i = 0 To k(lo)
cyc(i) = -1
For ii = 0 To cy(lo, i)
For z = 0 To hj(lo)
If (cycle(lo, i, ii) - tnumf) = hjj(lo, z) Then
cyc(i) = cyc(i) + 1
onlyone(i, cyc(i)) = hjj(lo, z)
End If
Next z
Next ii
Next i

For i = 0 To k(lo)
Nouse(i) = 0
Next i

u = -1
For i = 0 To k(lo)
nub = -1
For ii = 0 To cyc(i)
```

```
has = 0
For z = 0 To k(lo)
If (i <> z) And (Nouse(z) = 0) Then
For s = 0 To cyc(z)
If onlyone(i, ii) = onlyone(z, s) Then
nub = nub + 1
has = 1
Exit For
End If
Next s
If has = 1 Then
Exit For
End If
End If
Next z
Next ii
If nub < cyc(i) Then
u = u + 1
use(u) = i
Else
Nouse(i) = 1
End If
Next i

Call updatecycle(u, use())
u = -1
Noneuse = -1
For i = 0 To k(lo)
nub = 0
nubc = 0
```

```
For ii = 0 To cy(lo, i)
For z = 0 To hj(lo)
If (cycle(lo, i, ii) - tnumf) = hjj(lo, z) Then
has = 0
nub = nub + 1
If Noneuse = -1 Then
Noneuse = 0
oneuse(0) = hjj(lo, z)
Else
For s = 0 To Noneuse
If hjj(lo, z) = oneuse(s) Then
has = 1
nubc = nubc + 1
End If
Next s
If has = 0 Then
Noneuse = Noneuse + 1
oneuse(Noneuse) = hjj(lo, z)
End If
End If
End If
Next z
Next ii
If nub <> nubc Then
u = u + 1
use(u) = i
End If
Next i
Call updatecycle(u, use())
End Sub
```

```
Sub fcycle()

Dim i As Integer
Dim add As Integer
Dim size As Integer
Dim ii As Integer
Dim match As Integer
Dim X As Integer
Dim xx As Integer
Dim z As Integer
Dim sum(100) As Integer
Dim sumpair As Integer
Dim sumall As Integer
Dim u As Integer
Dim use(100) As Integer
Dim out As Integer
Dim s As Integer
Dim sumpuk As Integer
Dim zz As Integer
Dim zi As Integer
Dim za As Integer
Dim zb As Integer
Dim yes As Integer
Dim compare As Integer
Dim nub As Integer
Dim zg As Integer
Dim doo As Integer
out = 0

If ((a(lo) <> -1) And (a(lo) <> ((2 ^ num) - 1))) Then
```

```
For i = 0 To k(lo)
add = 0
size = -1
For ii = 0 To num - 1
If strnum(lo, n(lo), i, ii) = -1 Then
size = size + 1
cycle(lo, i, size) = 2 ^ (num - 1 - ii)
Else
add = add + (strnum(lo, n(lo), i, ii) * (2 ^ (num - 1 - ii)))
End If
Next ii

s = size

For X = 0 To s - 1
For xx = X + 1 To s
size = size + 1
cycle(lo, i, size) = cycle(lo, i, X) + cycle(lo, i, xx)
Next xx
Next X

For match = 1 To s - 1
For X = 0 To s - 1 - match
For xx = X + 1 To s - match
sumpair = cycle(lo, i, X)
For z = xx To xx + match - 1
sumpair = sumpair + cycle(lo, i, z)
Next z
```

```
For zz = z To s
size = size + 1
cycle(lo, i, size) = sumpair + cycle(lo, i, zz)
Next zz
Next xx
Next X
Next match

size = size + 1
cycle(lo, i, size) = 0

For X = 0 To size
cycle(lo, i, X) = cycle(lo, i, X) + add + tnumf
Next X

cy(lo, i) = size
Next i

If (k(lo) <> 0) And (k(lo) <> 1) Then

For i = 0 To k(lo)
sum(i) = 0

For ii = 0 To cy(lo, i)
sum(i) = sum(i) + cycle(lo, i, ii)
Next ii

Next i

sumall = 0
```

```
For i = 0 To hj(lo)
sumall = sumall + hjj(lo, i) + tnumf
Next i


For i = 0 To k(lo) - 1
sumpair = 0
u = -1
u = u + 1
use(u) = i
sumpair = sumpair + sum(i)
For ii = i + 1 To k(lo)
u = u + 1
use(u) = ii
If ((sumpair + sum(ii)) = sumall) Then
out = checkdo(use(), u)
End If
If out = 1 Then
Exit For
End If
u = u - 1
Next ii
If out = 1 Then
Exit For
End If
Next i

If out = 0 Then
For match = 1 To k(lo) - 1
For i = 0 To k(lo) - 1 - match
For xx = i + 1 To k(lo) - match + 1
u = -1
u = u + 1
use(u) = i
sumpair = sum(i)
For z = xx To xx + match - 1

u = u + 1
use(u) = z
sumpair = sumpair + sum(z)
Next z

For ii = z To k(lo)
u = u + 1
use(u) = ii

For zz = 0 To u - 1
For zi = zz + 1 To u
If cy(lo, zz) > cy(lo, zi) Then
compare = cy(lo, zi) + 1
Else
compare = cy(lo, zz) + 1
End If
nub = 0
For za = 0 To cy(lo, zz)
For zb = 0 To cy(lo, zi)
If cycle(lo, zz, za) = cycle(lo, zi, zb) Then
nub = nub + 1
End If
```

```
Next zb
Next za
If (nub > 0) And (nub < compare) Then
yes = 1
End If
Next zi
Next zz


If ((sumpair + sum(ii) = sumall) Or (yes = 1)) Then

If u <> k(lo) Then
out = checkdo(use(), u)
End If
End If
u = u - 1
yes = 0

If out = 1 Then
Exit For
End If
Next ii

If out = 0 Then
For ii = i + 1 To xx - 1
u = u + 1
use(u) = ii

For zz = 0 To u - 1
For zi = zz + 1 To u
```

```
If cy(lo, zz) > cy(lo, zi) Then
compare = cy(lo, zi) + 1
Else
compare = cy(lo, zz) + 1
End If
nub = 0
For za = 0 To cy(lo, zz)
For zb = 0 To cy(lo, zi)
If cycle(lo, zz, za) = cycle(lo, zi, zb) Then
nub = nub + 1
End If
Next zb
Next za
If (nub > 0) And (nub < compare) Then
yes = 1
End If
Next zi
Next zz


If ((sumpair + sum(ii) = sumall) Or (yes = 1)) Then
doo = 1
For zg = 0 To u - 1
For zz = zg + 1 To u
If use(zg) = use(zz) Then
doo = 0
End If
Next zz
Next zg
If (u <> k(lo)) And (doo = 1) Then
```

```
out = checkdo(use(), u)
End If
End If
u = u - 1
yes = 0

If out = 1 Then
Exit For
End If
Next ii
End If

If out = 1 Then
Exit For
End If
Next xx

If out = 1 Then
Exit For
End If
Next i

If out = 1 Then
Exit For
Else
sumpair = 0
u = -1
u = u + 1
use(u) = k(lo)
sumpair = sumpair + sum(k(lo))
```

```
For xx = 0 To match - 1
u = u + 1
use(u) = xx
sumpair = sumpair + sum(xx)
Next xx
For ii = 0 To match
u = u + 1
use(u) = ii

For zz = 0 To u - 1
For zi = zz + 1 To u
If cy(lo, zz) > cy(lo, zi) Then
compare = cy(lo, zi) + 1
Else
compare = cy(lo, zz) + 1
End If
nub = 0
For za = 0 To cy(lo, zz)
For zb = 0 To cy(lo, zi)
If cycle(lo, zz, za) = cycle(lo, zi, zb) Then
nub = nub + 1
End If
Next zb
Next za
If (nub > 0) And (nub < compare) Then
yes = 1
End If
Next zi
Next zz
```

```
If ((sumpair + sum(ii) = sumall) Or (yes = 1)) Then

If u <> k(lo) Then
out = checkdo(use(), u)
End If
End If
u = u - 1
yes = 0

If out = 1 Then
Exit For
End If
Next ii
End If


Next match
End If

End If

za = -1

If (k(lo) <> 0) Then
If (dc(lo) > -1) Then
checkdcare
End If
End If
```

```
For z = 0 To k(lo)
For zz = 0 To cy(lo, z)
za = za + 1
cycle(lo, k(lo) + 1, za) = cycle(lo, z, zz)
Next zz
Next z

cy(lo, k(lo) + 1) = za

End If


End Sub


Sub minform()


Dim i As Integer
Dim io As Integer
Dim ii As Integer
Dim relize As Integer
Dim dif As Integer
Dim same As Integer
Dim doo As Integer
Dim rou As Integer
Dim locate As Integer
Dim use(100) As Integer
Dim str() As Integer
Dim c As Integer
Dim stt(5) As Integer
Dim ar As Integer
```

```
n(lo) = 0
k(lo) = -1
dif = 0
same = 0

If (a(lo) = -1) Or (a(lo) = (2 ^ num) - 1) Then
relize = 0
Else
If a(lo) = 0 Then
relize = 0
k(lo) = k(lo) + 1
ar = arrays(lo, 0)
str = binary(ar)
For ii = 0 To num - 1
strnum(lo, n(lo), k(lo), ii) = str(ii)
Next ii
Else
relize = 1
End If
End If


If relize = 1 Then
For i = 0 To a(lo)
k(lo) = k(lo) + 1
ar = arrays(lo, i)
str = binary(ar)
For ii = 0 To num - 1
strnum(lo, n(lo), k(lo), ii) = str(ii)
```

```
Next ii
Next i
End If

Do While relize = 1

relize = 0

For io = 0 To k(lo) - 1
For i = io + 1 To k(lo)
dif = 0
For ii = 0 To num - 1
If strnum(lo, n(lo), io, ii) <> strnum(lo, n(lo), i, ii) Then
dif = dif + 1
End If
Next ii
If dif = 1 Then
relize = 1
Exit For
Else
relize = 0
End If
Next i
If relize = 1 Then
Exit For
End If
Next io

If relize = 1 Then
rou = k(lo)
```

```
k(lo) = -1
doo = 1

For i = 0 To rou
use(i) = 0
Next i

For io = 0 To rou - 1
For i = io + 1 To rou
dif = 0
For ii = 0 To num - 1
If strnum(lo, n(lo), io, ii) <> strnum(lo, n(lo), i, ii) Then
dif = dif + 1
locate = ii
End If
Next ii

If dif = 1 Then
For ii = 0 To num - 1
stt(ii) = strnum(lo, n(lo), i, ii)
Next ii
stt(locate) = -1

For c = 0 To k(lo)
same = 0
For ii = 0 To num - 1
If strnum(lo, n(lo) + 1, c, ii) = stt(ii) Then
same = same + 1
End If
Next ii
```

```
If same = num Then
doo = 0
Exit For
Else
doo = 1
End If
Next c

If doo = 1 Then
k(lo) = k(lo) + 1
For ii = 0 To num - 1
strnum(lo, n(lo) + 1, k(lo), ii) = strnum(lo, n(lo), i, ii)
Next ii
strnum(lo, n(lo) + 1, k(lo), locate) = -1
End If
use(i) = 1
use(io) = 1
End If

If dif = 0 Then
k(lo) = k(lo) + 1
For ii = 0 To num - 1
strnum(lo, n(lo) + 1, k(lo), ii) = strnum(lo, n(lo), i, ii)
Next ii
use(i) = 1
use(io) = 1
End If

Next i
Next io
```

```
For i = 0 To rou
If use(i) = 0 Then
k(lo) = k(lo) + 1
For ii = 0 To num - 1
strnum(lo, n(lo) + 1, k(lo), ii) = strnum(lo, n(lo), i, ii)
Next ii
End If

Next i

n(lo) = n(lo) + 1
End If

Loop

End Sub


Function binary(z As Integer) As Integer()
Dim gg As Integer
Dim kg As Integer
Dim strr() As Integer
ReDim strr(num)

gg = 2 ^ (num - 1)

Do While Not ((gg = 0) And (z = 0))
z = z - gg
If z >= 0 Then
```

```
strr(kg) = 1
kg = kg + 1
Else
strr(kg) = 0
kg = kg + 1
z = z + gg
End If
gg = gg / 2
Loop

binary = strr

End Function


Sub appear()
Select Case num
Case 2
Frame1(0).Visible = True
Frame1(0).Caption = unk(1) & " \ " & unk(0)
tnumf = 0
tnuml = 3
Case 3
Frame1(1).Visible = True
Frame1(1).Caption = unk(1) & " " & unk(2) & " \ " & unk(0)
tnumf = 4
tnuml = 11
Case 4
Frame1(2).Visible = True
Frame1(2).Caption = unk(2) & " " & unk(3) & " \ " & unk(0) & " " & unk(1)
```

```
tnumf = 12
tnuml = 27
Case 5
Frame1(2).Visible = True
Frame1(3).Visible = True
Frame1(2).Caption = unk(3) & " " & unk(4) & " \ " & unk(1) & " " & unk(2)
Frame1(3).Caption = unk(3) & " " & unk(4) & " \ " & unk(1) & " " & unk(2)
Label3(0).Visible = True
Label3(1).Visible = True
Label3(0).Caption = unk(0) & " = 0"
Label3(1).Caption = unk(0) & " = 1"
tnumf = 12
tnuml = 43
End Select
End Sub

Sub fmess()
Dim mess As String
Dim i As Integer
Dim ii As Integer
Dim aorhj As Integer

For i = 0 To 15
messp(lo, i) = ""
Next i

Select Case a(lo)
Case -1
If logic = "0" Then
messp(lo, 0) = "1"
```

```
Else
messp(lo, 0) = "0"
End If
k(lo) = k(lo) + 1
Case ((2 ^ num) - 1)
If dc(lo) <> (2 ^ num) - 1 Then
If logic = "0" Then
messp(lo, 0) = "0"
Else
messp(lo, 0) = "1"
End If
Else
If logic = "0" Then
messp(lo, 0) = "1"
Else
messp(lo, 0) = "0"
End If
k(lo) = 0
End If
k(lo) = k(lo) + 1
Case Else
Select Case logic
Case "0"
For i = 0 To k(lo)
For ii = 0 To num - 1

Select Case strnum(lo, n(lo), i, ii)
Case 0
If messp(lo, i) = "" Then
mess = mess & "("
Else
mess = mess & "+"
messp(lo, i) = messp(lo, i) & "+"
End If

mess = mess & unk(ii)
messp(lo, i) = messp(lo, i) & unk(ii)

Case 1
If messp(lo, i) = "" Then
mess = mess & "("
Else
mess = mess & "+"
messp(lo, i) = messp(lo, i) & "+"
End If
mess = mess & unk(ii) & "*"
messp(lo, i) = messp(lo, i) & unk(ii) & "*"

End Select

Next ii
mess = mess & ")"
Next i

Case "1"
For i = 0 To k(lo)
For ii = 0 To num - 1

Select Case strnum(lo, n(lo), i, ii)
Case 0
```

```
mess = mess & unk(ii) & "*"
messp(lo, i) = messp(lo, i) & unk(ii) & "*"
Case 1
mess = mess & unk(ii)
messp(lo, i) = messp(lo, i) & unk(ii)
End Select

Next ii
If i <> k(lo) Then
mess = mess & "+"
End If
Next i

End Select
messp(lo, k(lo) + 1) = mess
End Select

End Sub

Sub initail()
Dim i As Integer
Dim j As String
Dim lis As Integer
Dim ii As Integer
Dim aorhj As Integer

lis = List2.ListIndex
If dc(lis) > -1 Then
aorhj = hj(lis)
Else
```

```
aorhj = a(lis)
End If



For i = 0 To aorhj
Label1(arrays(lis, i) + tnumf).Caption = logic
Next i

For i = 0 To dc(lis)
Label1(dcare(lis, i)).Caption = "d"
Next i

If k(lis) = 0 Then
Label5.Caption = messp(lis, 0)
If (aorhj = ((2 ^ num) - 1)) Or (a(lis) = ((2 ^ num) - 1)) Then
For i = tnumf To tnuml
Label1(i).BackColor = QBColor(10)
Next i
End If

If (aorhj <> -1) And (aorhj <> (2 ^ num) - 1) Then
For i = 0 To cy(lis, k(lis))
Label1(cycle(lis, k(lis), i)).BackColor = QBColor(10)
Next i
End If

Else

If (aorhj <> -1) And (aorhj <> (2 ^ num) - 1) Then
For i = 0 To k(lis)
```

```
List1.AddItem messp(lis, i)
Next i
j = messp(lis, k(lis) + 1) & " (ALL)"
List1.AddItem j
Label5.Caption = messp(lis, k(lis) + 1)
For i = 0 To cy(lis, k(lis) + 1)
Label1(cycle(lis, k(lis) + 1, i)).BackColor = QBColor(10)
Next i

Else
Label5.Caption = messp(lis, 0)
End If

End If

If k(lis) > 0 Then
Command4.Visible = True
End If
If dc(lis) = (2 ^ num) - 1 Then
Command4.Visible = False
End If

If Len(Label5.Caption) > 40 Then
Label5.Width = 7000
Label5.Font.size = 8
List1.Width = 7000
End If

End Sub
```

```
Sub fthree()
Dim arr() As Integer
Dim dcar() As Integer
Dim i As Integer
Dim ii As Integer

For i = 0 To numout - 1
arr = Form3.continlogic(i)
a(i) = Form3.aa
dcar = Form3.contindc(i)
dc(i) = Form3.dcc
For ii = 0 To a(i)
arrays(i, ii) = arr(ii)
Next ii
For ii = 0 To dc(i)
dcare(i, ii) = dcar(ii)
Next ii
Next i
End Sub

Sub ffour()
Dim arr() As Integer
Dim dcar() As Integer
Dim i As Integer
Dim ii As Integer
Dim sta As String
Dim std As String
Dim X As Integer

For i = 0 To numout - 1
```

```
sta = Form4.Text1(i).Text
arr = Form4.cutnuml(sta)
a(i) = Form4.a
std = Form4.Text2(i).Text
dcar = Form4.cutnumd(std)
dc(i) = Form4.dc
For ii = 0 To a(i)
arrays(i, ii) = arr(ii)
Next ii
For ii = 0 To dc(i)
dcare(i, ii) = dcar(ii) + tnumf
Next ii

Next i

End Sub

Sub ffive()
Dim i As Integer
Dim v As Integer
Dim ii As Integer

For i = 0 To numout - 1
v = i
a(i) = -1
dc(i) = -1
For ii = 0 To (2 ^ num) - 1
Select Case Form5.Label2(v).Caption
Case logic
a(i) = a(i) + 1
```

```
arrays(i, a(i)) = ii
Case ""
dc(i) = dc(i) + 1
dcare(i, dc(i)) = ii + tnumf
End Select
v = v + 4
Next ii
Next i

End Sub

Function fnine(numm As Integer, nummo As Integer)
Dim X As Integer
Dim i As Integer
Dim ii As Integer
Dim lon As Integer
Dim Y As Integer
Dim char As String
Dim add As Integer
Dim addc As Integer
Dim g As Integer
Dim vk As Integer
Dim m As Integer
Dim j(32) As String

a(g) = -1
dc(g) = -1

lon = Len(Form9.Label10(0).Caption)
```

```
For X = 0 To outf - 1
addc = 0

For Y = 1 To lon
vk = 0
m = -1

For i = 0 To (2 ^ numm) - 1
For ii = 0 To (2 ^ outf) - 1
m = m + 1
If Y = 1 Then
j(m) = Form9.Label10(ii + add + addc).Caption
char = Mid$(j(m), Y, 1)
Else
char = Mid$(j(m), Y, 1)
End If

Select Case char
Case logic
a(g) = a(g) + 1
arrays(g, a(g)) = vk
Case "d"
dc(g) = dc(g) + 1
dcare(g, dc(g)) = vk + tnumf

Case ""
dc(g) = dc(g) + 1
dcare(g, dc(g)) = vk + tnumf

End Select
```

```
vk = vk + 1
Next ii
addc = addc + 24
Next i
g = g + 1
a(g) = -1
dc(g) = -1
Next Y
add = add + 8
Next X
If Form1.Option4.value = True Then
addc = 0
add = 0
vk = 0

For X = 0 To nummo - 1
addc = 0
vk = 0
For i = 0 To (2 ^ numm) - 1
add = 0
For ii = 0 To (2 ^ outf) - 1
char = Form9.Label3(X + add + addc).Caption

Select Case char
Case logic
a(g) = a(g) + 1
arrays(g, a(g)) = vk

Case "-"
dc(g) = dc(g) + 1
```

```
dcare(g, dc(g)) = vk + tnumf

End Select
vk = vk + 1
add = add + 3
Next ii
addc = addc + 24
Next i
g = g + 1
a(g) = -1
dc(g) = -1
Next X
End If

End Function

Sub findff()
Dim i As Integer
Dim X As String
Dim b As Integer

For i = 1 To 3
If (2 ^ i) >= sta Then
outf = i
Exit For
End If
Next i

For i = 0 To outf - 1
outff(i) = "Q" & str$(outf - 1 - i)
```

```
Next i

Select Case Form9.typeff
Case 0
X = "D"
Case 1
X = "JK"
Case 2
X = "T"
End Select

If X = "JK" Then
For i = 0 To outf - 1
inff(b) = "J" & str$(outf - 1 - i)
b = b + 1
inff(b) = "K" & str$(outf - 1 - i)
b = b + 1
Next i
inf = b

Else
For i = 0 To outf - 1
inff(i) = X & str$(outf - 1 - i)
Next i
inf = outf
End If

End Sub
```

```
Private Sub Command1_Click()
Load Form10
Form10.Show
Form6.Visible = False
End Sub

Private Sub Command2_Click()
If Form1.Option1 = True Then
Form3.Visible = True
End If
If Form1.Option2 = True Then
Form4.Visible = True
End If
If Form1.Option3 = True Then
Form5.Visible = True
End If
If Form1.Option4 = True Then
Form9.Visible = True
End If
Unload Form6
End Sub

Private Sub Command3_Click()
End
End Sub

Private Sub Command4_Click()
If List1.Visible = True Then
ListI.Visible = False
Else
```

```
List1.Visible = True
End If
End Sub

Private Sub Form_DragDrop(Source As Control, X As Single, Y As Single)
Frame1(frameindex).Top = Y - firsty
Frame1(frameindex).Left = X - firstx
xy = 0
End Sub

Private Sub Form_Load()
Dim i As Integer
Dim ii As Integer
Dim X As Integer
Dim numm As Integer
Dim ss(15) As String
Dim nummo As Integer
Dim Y As Integer
Dim z As Integer
Dim g As Integer
Dim za As Integer
Dim zz As Integer
Dim j As Integer

If (Form1.Option4.value = True) Or (Form1.Option5.value = True) Then

If Form7.Combo1.Text = "Minterm" Then
logic = "1"
Else
logic = "0"
```

```
End If

unk = Form7.cuts(Form7.Text1.Text)
num = Form7.num
output = Form7.cuts(Form7.Text2.Text)
numout = Form7.numout
State = Form7.cuts(Form7.Text3.Text)
sta = Form7.sta
findff

numm = num
nummo = numout
For i = num To num + outf - 1
unk(i) = outff(i - num)
Next i
num = num + outf

If Form1.Option4.value = True Then
For i = 0 To inf - 1
ss(i) = inff(i)
Next i
For i = inf To numout + inf - 1
ss(i) = output(i - inf)
Next i
numout = numout + inf
For i = 0 To numout - 1
output(i) = ss(i)
Next i
Else
For i = 0 To inf - 1
```

```
output(i) = inff(i)
Next i
numout = inf
Command1.Visible = True
End If

appear

Call fnine(numm, nummo)

Else

If Form2.Combo1.Text = "Minterm" Then
logic = "1"
Else
logic = "0"
End If

unk = Form2.cuts(Form2.Text1.Text)
num = Form2.num
output = Form2.cuts(Form2.Text2.Text)
numout = Form2.numout

appear

If Form1.Option1 = True Then
fthree
End If
If Form1.Option2 = True Then
ffour
```

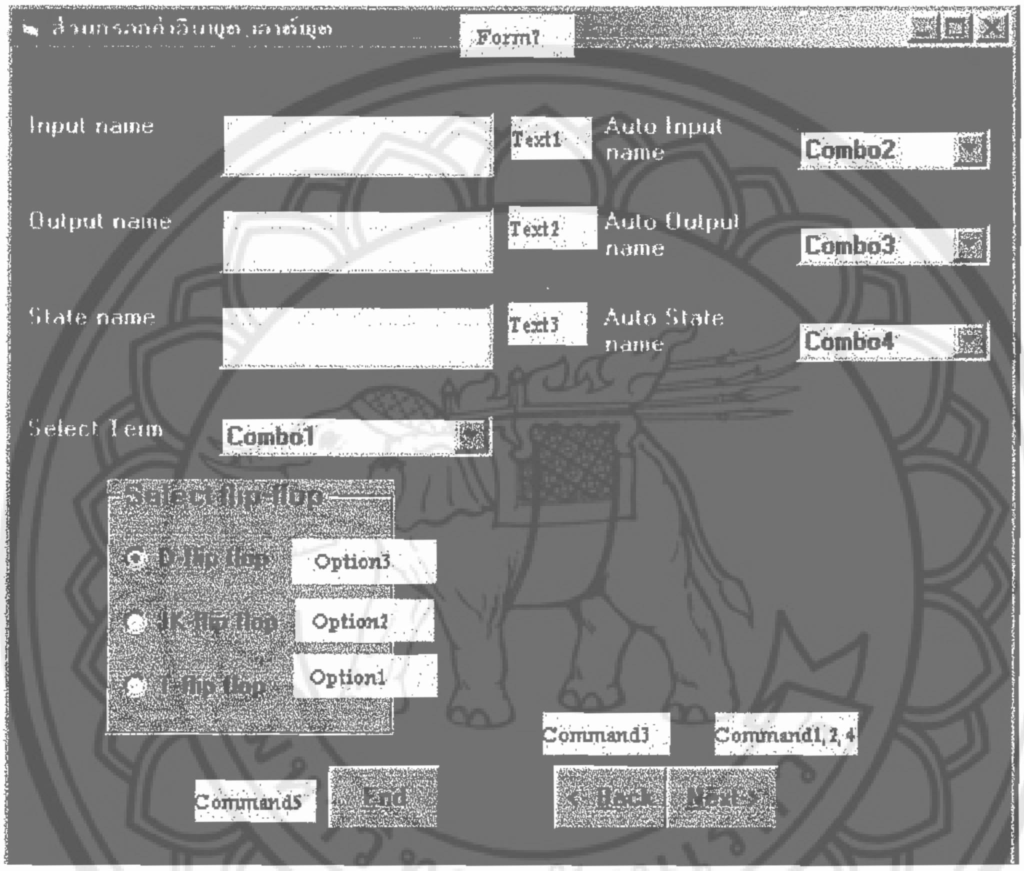
```
End If
If Form1.Option3 = True Then
ffive
End If

End If



For i = 0 To numout - 1
For ii = 0 To a(i)
hjj(i, ii) = arrays(i, ii)
Next ii
hj(i) = a(i)
Next i

For i = 0 To numout - 1
X = a(i)
For ii = 0 To dc(i)
X = X + 1
arrays(i, X) = dcare(i, ii) - tnumf
Next ii
a(i) = X
Next i

For lo = 0 To numout - 1
minform
fcycle
Next lo
```

```
For lo = 0 To numout - 1
fmess
Next lo

For i = 0 To numout - 1
List2.AddItem output(i)
Next i

List2.ListIndex = 0
oldlist = 0

End Sub


Private Sub Frame1_DragOver(Index As Integer, Source As Control, X As Single, Y As Single,
State As Integer)

If xy = 0 Then
frameindex = Index
firstx = X
firsty = Y
xy = 1
End If
End Sub

Private Sub List1_Click()
Dim i As Integer
Dim lis As Integer

lis = List2.ListIndex
```

```
Label5.Caption = messp(lis, List1.ListIndex)

For i = tnumf To tnuml
Label1(i).BackColor = QBColor(15)
Next i

For i = 0 To cy(lis, List1.ListIndex)
Label1(cycle(lis, List1.ListIndex, i)).BackColor = QBColor(10)
Next i

List1.Visible = False
End Sub

Private Sub List2_Click()
Dim i As Integer

For i = tnumf To tnuml
Label1(i).Caption = ""
Label1(i).BackColor = QBColor(15)
Next i

If List1.ListCount > 0 Then
For i = 0 To k(oldlist) + 1
List1.RemoveItem k(oldlist) + 1 - i
Next i
End If
Command4.Visible = False
oldlist = List2.ListIndex

initail
```

```
End Sub
```

Form7 (ส่วนการกรอกค่าอินพุต,เอาท์พุต)



```
Public numout As Integer
Public num As Integer
Public a As Integer
Dim ftextok As Integer
Dim stextok As Integer
Dim combook As Integer
Dim inp As Integer
Public sta As Integer
Dim valueok As Integer
```

```
Public Function cuts(st As String) As String()
Dim lon As Integer
Dim i As Integer
Dim ii As Integer
Dim numm As Integer
Dim puk As String
Dim va As String
Dim kk As String
Dim repeat As Integer
Dim unk() As String
ReDim unk(20)
repeat = 0
ftextok = 0
numm = 0
kk = ""
lon = Len(st)
If lon = 0 Then
If Not (st Like Text1.Text) Then
MsgBox ("คุณต้องใส่ตัวแปร")
Exit Function
End If
Else
For i = 1 To lon
puk = Left$(st, i)
va = Right$(puk, 1)
If va = "," Then
If kk <> "" Then
unk(numm) = kk
numm = numm + 1
kk = ""
```

```
Else
MsgBox ("คุณไม่ใส่ตัวแปรหรือผิดหลักโปรแกรม")
Exit Function
End If
Else
If va <> " " Then
If Not ((va >= "0") And (va <= "9")) Then
kk = kk & va
Else
MsgBox ("you should not define input,output is number")
Exit Function
End If
End If
End If
Next i
End If

If kk <> "" Then
unk(numm) = kk
numm = numm + 1
End If

If st Like Text1.Text Then
If numm > 3 Then
MsgBox ("you have input 0 to 3")
Exit Function
End If
End If

If st Like Text3.Text Then
If numm > 8 Then
MsgBox ("you have state 1 to 8")
Exit Function
End If
End If

If st Like Text2.Text Then
If numm > 3 Then
MsgBox ("you have output 1 to 3")
Exit Function
End If
End If

For i = 0 To numm - 2
For ii = i + 1 To numm - 1
If unk(i) = unk(ii) Then
repeat = 1
End If
Next ii
Next i

If repeat = 1 Then
MsgBox ("you can not put same input or output")
Exit Function
End If
If st Like Text1.Text Then
num = numm
Else
numout = numm
End If
```

```
ftextok = 1
If st Like Text1.Text Then
inp = numm
End If
If st Like Text3.Text Then
sta = numm
End If
cuts = unk

End Function

Sub checkvalue()
Dim s As Integer
Dim i As Integer

valueok = 0

For i = 1 To 3
If sta <= (2 ^ i) Then
s = i
Exit For
End If
Next i
If ((s + inp) = 1) Or ((s + inp) > 5) Then
MsgBox ("input add state(flip-flop) must have 2 to 5")
Exit Sub
End If

valueok = 1
```

```
End Sub

Sub checkcombo()
combook = 0
If Combo1.Text <> "" Then
combook = 1
Else
MsgBox ("you must choose term")
End If
End Sub

Private Sub Combo2_Click()
Select Case Combo2.Text
Case "1"
Text1.Text = "S"
Case "2"
Text1.Text = "S,T"
Case "3"
Text1.Text = "S,T,U"
Case "4"
Text1.Text = "S,T,U,V"
End Select
End Sub

Private Sub Combo3_Click()
Select Case Combo3.Text
Case "1"
Text2.Text = "Z"
Case "2"
Text2.Text = "Y,Z"
```

```
Case "3"
Text2.Text = "X,Y,Z"
Case "4"
Text2.Text = "W,X,Y,Z"
End Select
End Sub

Private Sub Command1_Click()
Dim un() As String
un = cuts(Text1.Text)

If ftextok = 1 Then
checkcombo
End If

If (ftextok = 1) And (combook = 1) Then
un = cuts(Text2.Text)

If ftextok = 1 Then
checkcombo
End If

If (ftextok = 1) And (combook = 1) Then
Load Form3
Form3.Show
Form2.Visible = False
End If

End If
End Sub
```

```
Private Sub Command2_Click()
Dim un() As String
Dim i As Integer

un = cuts(Text1.Text)

If ftextok = 1 Then
checkcombo
End If

If (ftextok = 1) And (combook = 1) Then
un = cuts(Text2.Text)

If ftextok = 1 Then
checkcombo
End If

If (ftextok = 1) And (combook = 1) Then
Load Form8
Form8.Show
Form2.Visible = False
End If
End If

For i = 0 To numout - 1
Form4.Label1(i).Visible = True
Form4.Label2(i).Visible = True
Form4.Text1(i).Visible = True
Form4.Text2(i).Visible = True
```

```
Form4.Label1(i).Caption = un(i)
Next i

End Sub

Private Sub Combo4_Click()
Dim g As Integer
Dim i As Integer

Text3.Text = ""

g = Asc("A")
g = g - 1
For i = 0 To Combo4.ListIndex
g = g + 1
If i = Combo4.ListIndex Then
Text3.Text = Text3.Text & Chr$(g)
Else
Text3.Text = Text3.Text & Chr$(g) & ","
End If
Next i
End Sub

Private Sub Command3_Click()
Unload Form1
Load Form1
Form1.Show
Unload Form7
End Sub
```

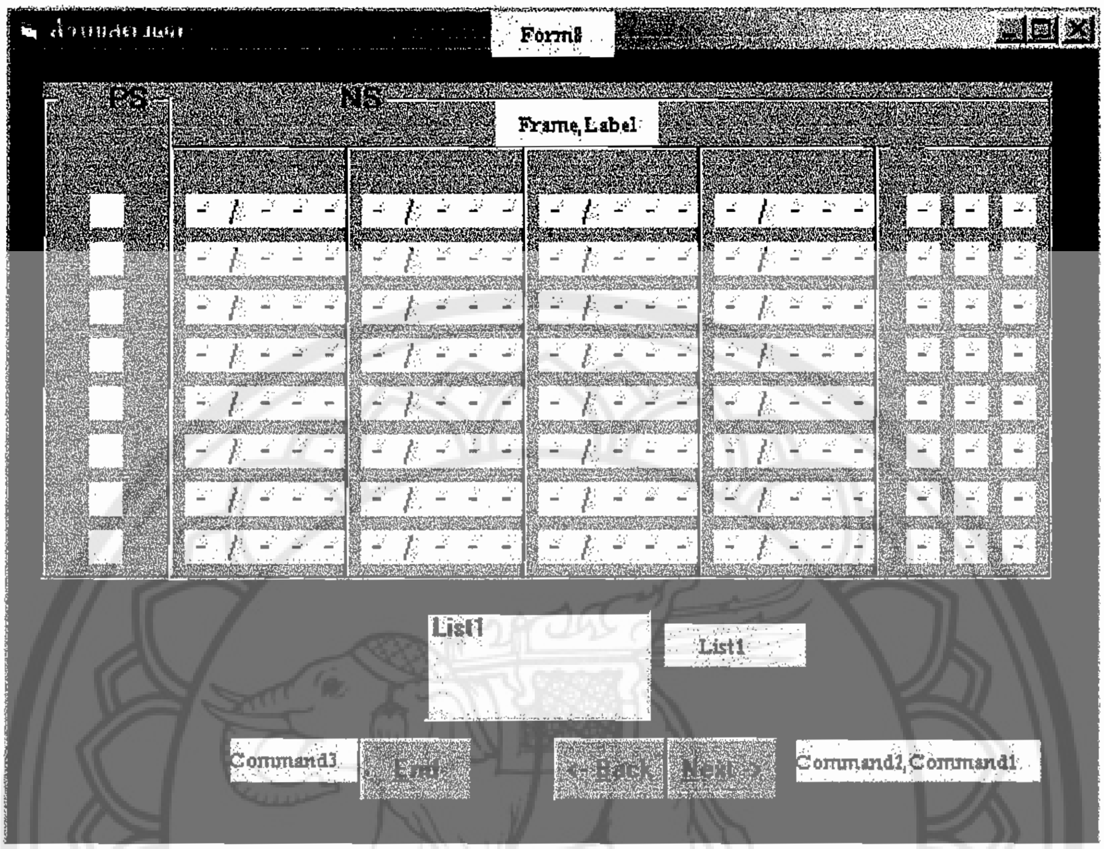
```
Private Sub Command4_Click()
Dim inn() As String
Dim oun() As String
Dim stn() As String
Dim i As Integer
Dim ii As Integer
Dim dc As Integer
Dim dcar() As Integer

If Text1.Text = "" Then
i = i + 1
End If

If Text2.Text = "" Then
i = i + 1
End If

If Text3.Text = "" Then
i = i + 1
End If

If i > 1 Then
MsgBox ("you must put value")
Exit Sub
End If

inn = cuts(Text1.Text)

If ftextok = 1 Then
oun = cuts(Text2.Text)
```

```
End If

If ftextok = 1 Then
stn = cuts(Text3.Text)
End If
checkcombo
If (ftextok = 1) And (combook = 1) Then
checkvalue
If valueok = 1 Then
For i = 0 To num - 1
For ii = 0 To numout - 1
If inn(i) Like oun(ii) Then
MsgBox ("inputname and outputname has same")
Exit Sub
End If
Next ii
For ii = 0 To sta - 1
If inn(i) Like stn(ii) Then
MsgBox ("inputname and statename has same")
Exit Sub
End If
Next ii
Next i

For i = 0 To numout - 1
For ii = 0 To sta - 1
If oun(i) Like stn(ii) Then
MsgBox ("outputname and statename has same")
Exit Sub
End If
```

```
Next ii
Next i
Load Form8
Form8.Show
Form7.Visible = False
End If
End If
End Sub
Private Sub Command5_Click()
End
End Sub
Private Sub Form_Load()
num = 0
numout = 0
End Sub
```

Form8 (ส่วนแสดงผล)

```
Dim numout As Integer
Dim output() As String
Dim unk() As String
Dim num As Integer
Dim State() As String
Dim sta As Integer
```



```
Function binary(z As Integer) As Integer()
Dim gg As Integer
Dim kg As Integer
Dim strr() As Integer
Dim numout As Integer
Dim output() As String
Dim unk() As String
Dim num As Integer
Dim State() As String
Dim sta As Integer
Function binary(z As Integer) As Integer()
Dim gg As Integer
Dim kg As Integer
Dim strr() As Integer
Dim numout As Integer
Dim output() As String
Dim unk() As String
Dim num As Integer
Dim State() As String
Dim sta As Integer
Function binary(z As Integer) As Integer()
Dim gg As Integer
Dim kg As Integer
Dim strr() As Integer
ReDim strr(num)
gg = 2 ^ (num - 1)
Do While Not ((gg = 0) And (z = 0))
z = z - gg
If z >= 0 Then
strr(kg) = 1
kg = kg + 1


Else
strr(kg) = 0
kg = kg + 1
z = z + gg
End If
gg = gg / 2
Loop
binary = strr
End Function

Private Sub Command1_Click()
Form8.Visible = False
```

```
Function binary(z As Integer) As Integer()
Dim gg As Integer
Dim kg As Integer
Dim strr() As Integer
ReDim strr(num)
gg = 2 ^ (num - 1)
Do While Not ((gg = 0) And (z = 0))
z = z - gg
If z >= 0 Then
strr(kg) = 1
kg = kg + 1


Else
strr(kg) = 0
kg = kg + 1
z = z + gg
End If
gg = gg / 2
Loop
binary = strr
End Function

Private Sub Command1_Click()
Form8.Visible = False
Load Form9
Form9.Show
End Sub

Private Sub Command2_Click()
```

```
Unload Form7
Load Form7
Form7.Show
Unload Form8
End Sub

Private Sub Command3_Click()
End
End Sub

Private Sub Form_Load()
Dim i As Integer
Dim ii As Integer
Dim X As String
Dim bi() As Integer
Dim Y As String
Dim value As Integer
Dim shift As Integer
Dim bb As Integer
Dim h As Integer

unk = Form7.cuts(Form7.Text1.Text)
num = Form7.num
output = Form7.cuts(Form7.Text2.Text)
numout = Form7.numout
State = Form7.cuts(Form7.Text3.Text)
sta = Form7.sta

For ii = 0 To numout - 1
For i = 0 To sta - 1
```

```
Label5(i + h).Visible = True
Label5(i + h + 24).Visible = True
Next i
h = h + 8
Next ii

For i = 0 To sta - 1
Label1(i).Visible = True
Label1(i).Caption = State(i)
Label1(i + 8).Visible = True
Label1(i + 8).Caption = State(i)
Next i

If Form1.Option4.value = True Then
Frame1(0).Caption = Frame1(0).Caption & "/"
Frame1(1).Caption = Frame1(1).Caption & "/"
For i = 0 To numout - 1
Frame1(0).Caption = Frame1(0).Caption & output(i)
Frame1(1).Caption = Frame1(1).Caption & output(i)
Next i
End If

For i = 0 To num - 1
X = X & unk(i)
Next i

If num <> 0 Then
For i = 0 To (2 ^ num) - 1
Frame3(i).Visible = True
Frame3(i).Caption = "   " & X & "= "
```

```
bb = i
bi = binary(bb)
For ii = 0 To num - 1
Y = str$(bi(ii))
Frame3(i).Caption = Frame3(i).Caption & Y
Next ii
Next i
Else
Frame3(0).Visible = True
End If

If num < 3 Then
Frame1(0).Width = Frame1(0).Width - (1320 * (5 - (2 ^ num)))
Frame4(0).Left = Frame4(0).Left - (1320 * (4 - (2 ^ num)))
End If

If num = 1 Then
Frame1(0).Caption = "            " & Frame1(0).Caption
End If
If (num = 2) Or (num = 3) Then
Frame1(0).Caption = "                                      " & Frame1(0).Caption
End If
value = sta * (2 ^ num)

For i = 0 To numout - 1
add = 0
shift = 0
bb = 0
For ii = 0 To value - 1
```

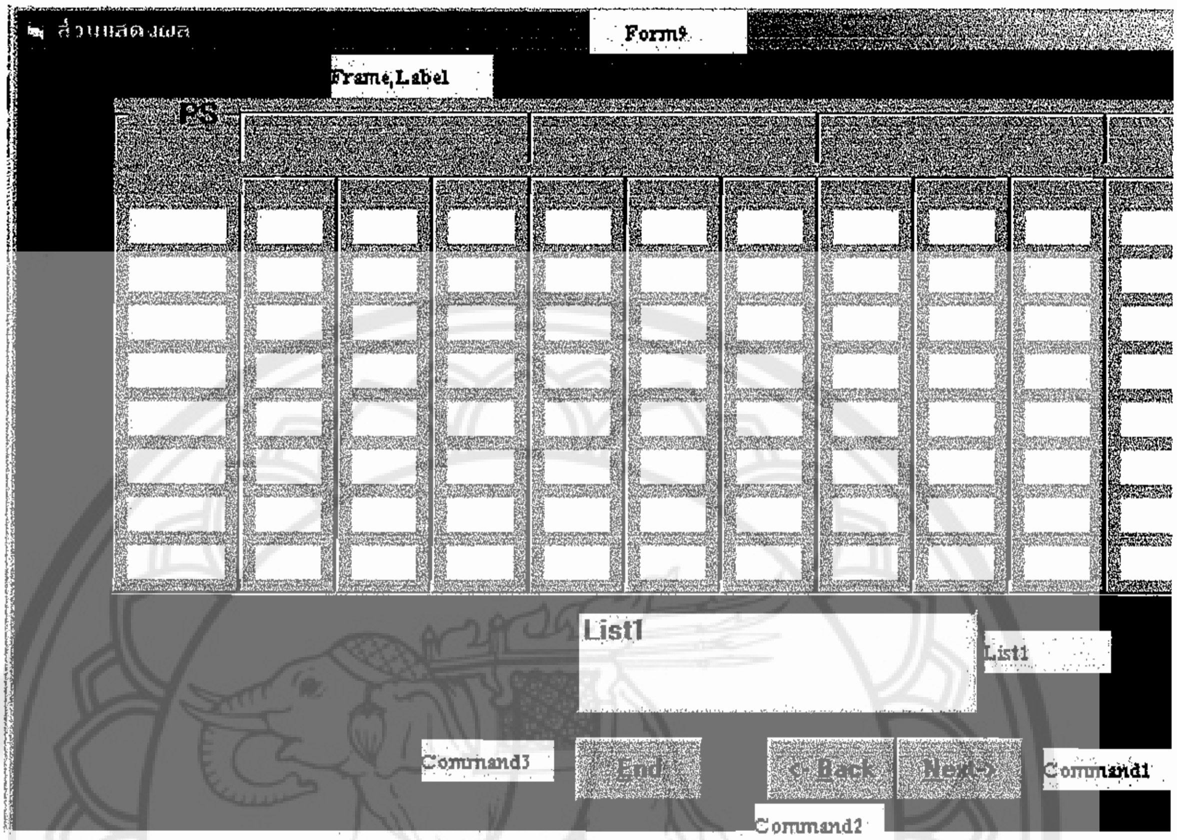
```
If bb = sta Then
shift = shift + 8
bb = 0
add = 0
End If
Label2(bb + shift).Visible = True
If Form1.Option4.value = True Then
Label4(bb + shift).Visible = True
Label3(i + add + (3 * shift)).Visible = True
Else
Label2(bb + shift).Left = 480
End If
add = add + 3
bb = bb + 1
Next ii
Next i
If num < 3 Then
List1.AddItem "State Table"
Else
List1.AddItem "State Table1"
List1.AddItem "State Table2"
End If

If Form1.Option5.value = True Then
If num < 3 Then
Frame1(0).Width = Frame1(0).Width + 1320
End If
For i = 0 To numout - 1
Frame4(0).Caption = Frame4(0).Caption & output(i)
```

```
Frame4(1).Caption = Frame4(1).Caption & output(i)
Next i
End If
List1.ListIndex = 0
End Sub

Private Sub Label2_Click(Index As Integer)
Select Case Label2(Index).Caption
Case State(sta - 1)
Label2(Index).Caption = "-"
Case "-"
Label2(Index).Caption = State(0)
Case Else
For i = 0 To sta - 2
If State(i) = Label2(Index).Caption Then
Label2(Index).Caption = State(i + 1)
Exit For
End If
Next i
End Select
End Sub

Private Sub Label3_Click(Index As Integer)
Select Case Label3(Index).Caption
Case "-"
Label3(Index).Caption = "0"
Case "1"
Label3(Index).Caption = "-"
Case "0"
Label3(Index).Caption = "1"
```

```
End Select
End Sub

Private Sub Label5_Click(Index As Integer)

Select Case Label5(Index).Caption
Case "-"
Label5(Index).Caption = "0"
Case "1"
Label5(Index).Caption = "-"
Case "0"
Label5(Index).Caption = "1"
End Select
End Sub

Private Sub List1_Click()
Dim i As Integer
For i = 0 To 1
Frame1(i).Visible = False
Next i
Frame1(List1.ListIndex).Visible = True
End Sub
```

Form9 (ส่วนแสดงผล)

```
Dim numout As Integer
Dim output() As String
Dim unk() As String
Dim num As Integer
Dim State() As String
```



```
Dim sta As Integer
Dim outff(5) As String
Dim outf As Integer
Dim inff(10) As String
Dim inf As Integer
Public typeff As Integer
Dim flipflop(3, 2, 2) As String

Function stonum(st As String, z As Integer) As String
Dim lon As Integer
Dim va As String
```

```
va = Left$(st, z)
va = Right$(va, 1)

stonum = va
End Function

Function binary(z As Integer) As Integer()
Dim gg As Integer
Dim kg As Integer
Dim strr() As Integer
ReDim strr(outf)

gg = 2 ^ (outf - 1)

Do While Not ((gg = 0) And (z = 0))
z = z - gg
If z >= 0 Then
strr(kg) = 1
kg = kg + 1
Else
strr(kg) = 0
kg = kg + 1
z = z + gg
End If
gg = gg / 2
Loop

binary = strr

End Function
```

```
Sub stassign()
Dim i As Integer
Dim bb As Integer
Dim bi() As Integer
Dim Y As String

For i = 0 To outf - 1
Frame4.Caption = Frame4.Caption & outff(i)
Next i

For i = 0 To sta - 1
Label5(i).Visible = True
Label6(i).Visible = True
Label5(i).Caption = State(i)
Label6(i).Caption = ""
bb = i
bi = binary(bb)
For ii = 0 To outf - 1
Y = str$(bi(ii))
Label6(i).Caption = Label6(i).Caption & Y
Next ii
Next i

End Sub

Sub findff()
Dim i As Integer
Dim X As String
Dim b As Integer
```

```
For i = 1 To 3
If (2 ^ i) >= sta Then
outf = i
Exit For
End If
Next i

For i = 0 To outf - 1
outff(i) = "Q" & str$(outf - 1 - i)
Next i

Select Case typeff
Case 0
X = "D"
Case 1
X = "JK"
Case 2
X = "T"
End Select

If X = "JK" Then
For i = 0 To outf - 1
inff(b) = "J" & str$(outf - 1 - i)
b = b + 1
inff(b) = "K" & str$(outf - 1 - i)
b = b + 1
Next i
inf = b
```

```
Else
For i = 0 To outf - 1
inff(i) = X & str$(outf - 1 - i)
Next i
inf = outf
End If

End Sub

Sub trantable()
Dim i As Integer
Dim outx As String
Dim numx As String
Dim X As String
Dim bb As Integer
Dim bi() As Integer
Dim add As Integer
Dim Y As String
Dim dd As Integer
Dim h As Integer

For ii = 0 To numout - 1
For i = 0 To sta - 1
Label11(i + h).Caption = Form8.Label5(i + h).Caption
Label11(i + h + 24).Caption = Form8.Label5(i + h + 24).Caption
Label11(i + h).Visible = True
Label11(i + h + 24).Visible = True
Next i
h = h + 8
Next ii
```

```
For i = 0 To outf - 1
outx = outx & outff(i)
Next i
For i = 0 To numout - 1
numx = numx & output(i)
Next i
For i = 0 To num - 1
X = X & unk(i)
Next i

Label7(0).Caption = outx
Label7(1).Caption = Label7(1).Caption & outx

For i = 0 To sta - 1
Label1(i).Caption = Label6(i).Caption
Label1(i + 8).Caption = Label6(i).Caption
Label1(i).Visible = True
Label1(i + 8).Visible = True
Next i

If num = 3 Then
dd = 2
Else
dd = num
End If

For i = 0 To (2 ^ dd) - 1

For bb = 0 To sta - 1
```

```
Label2(add + bb).Visible = True
Label2(add + bb + 32).Visible = True
If Form1.Option4.value = True Then
Label4(add + bb).Visible = True
Label4(add + bb + 32).Visible = True
Else
Label2(add + bb).Left = 480
End If
Next bb
add = add + 8
Next i
Frame1(0).Caption = Frame1(0).Caption & "(" & outx & ")"
Frame1(1).Caption = Frame1(1).Caption & "(" & outx & ")"

If Form1.Option4.value = True Then
Frame1(0).Caption = Frame1(0).Caption & "/" & numx
Frame1(1).Caption = Frame1(1).Caption & "/" & numx
End If

If num < 3 Then
Frame1(0).Width = Frame1(0).Width - (1560 * (5 - (2 ^ num)))
Frame9(0).Left = Frame9(0).Left - (1560 * (4 - (2 ^ num)))
End If

If num = 1 Then
Frame1(0).Caption = "          " & Frame1(0).Caption
End If
If (num = 2) Or (num = 3) Then
Frame1(0).Caption = "                              " & Frame1(0).Caption
End If
```

```
If num <> 0 Then
For i = 0 To (2 ^ num) - 1
Frame3(i).Visible = True
Frame3(i).Caption = "   " & X & "= "

bb = i
bi = Form8.binary(bb)
For ii = 0 To num - 1
Y = str$(bi(ii))
Frame3(i).Caption = Frame3(i).Caption & Y
Next ii
Next i
Else
Frame3(0).Visible = True
End If

For i = 0 To 191
Label3(i).Visible = Form8.Label3(i).Visible
Label3(i).Caption = Form8.Label3(i).Caption
Next i

For i = 0 To 63
If Form8.Label2(i).Caption = "-" Then
Label2(i).Caption = "-"
Else
For bb = 0 To sta - 1
If Form8.Label2(i).Caption = State(bb) Then
Label2(i).Caption = Label6(bb).Caption
Exit For
```

```
End If
Next bb
End If
Next i

If Form1.Option5.value = True Then
If num < 3 Then
Frame1(0).Width = Frame1(0).Width + 1560
End If

Frame9(0).Caption = Frame9(0).Caption & numx
Frame9(1).Caption = Frame9(1).Caption & numx
End If
End Sub

Sub excitable()
Dim i As Integer
Dim outx As String
Dim numx As String
Dim X As String
Dim bb As Integer
Dim bi() As Integer
Dim cc As Integer
Dim Y As String
Dim value As Integer
Dim ii As Integer
Dim dd As Integer
Dim add As Integer
Dim addc As Integer
Dim addt As Integer
Dim addce As Integer
Dim adde As Integer
Dim PS As Integer
Dim NS As Integer
Dim PSS As String
Dim NSS As String

For i = 0 To outf - 1
outx = outx & outff(i)
Next i
For i = 0 To numout - 1
numx = numx & output(i)
Next i
For i = 0 To num - 1
X = X & unk(i)
Next i

For i = 0 To sta - 1
Label8(i).Caption = Label6(i).Caption
Label8(i + 8).Caption = Label6(i).Caption
Label8(i).Visible = True
Label8(i + 8).Visible = True
Next i
Label9(0).Caption = outx
Label9(1).Caption = Label9(1).Caption & outx

If num = 3 Then
dd = 2
Else
dd = num
```

```
End If

For i = 0 To (2 ^ dd) - 1
add = 0
For ii = 0 To outf - 1
For bb = 0 To sta - 1
Label10(add + addc + bb).Visible = True
Label10(add + addc + bb + 96).Visible = True
Next bb
add = add + 8
Next ii
addc = addc + 24
Next i

If num < 3 Then
Frame5(1).Width = Frame5(1).Width - (2160 * (4 - (2 ^ num)))
End If
ii = 1
If outf < 3 Then

For i = 0 To (2 ^ dd) - 1
Frame7(i).Width = Frame7(i).Width - ((3 - outf) * 720)
If (i + 1) <= 3 Then
Frame7(i + 1).Left = Frame7(i + 1).Left - (ii * ((3 - outf) * 720))
End If
ii = ii + 1
Next i
Frame5(1).Width = Frame5(1).Width - ((((3 - outf) * (2 ^ dd)) * 720))
ii = 1
For i = 4 To 7
```

```
Frame7(i).Width = Frame7(i).Width - ((3 - outf) * 720)
If (i + 1) <= 7 Then
Frame7(i + 1).Left = Frame7(i + 1).Left - (ii * ((3 - outf) * 720))
End If
ii = ii + 1
Next i
Frame5(0).Width = Frame5(0).Width - ((((3 - outf) * (2 ^ dd)) * 720))
End If

Select Case outf
Case 1
X = X & "="
Case 2
X = "      " & X & "= "
Case 3
X = "         " & X & "= "
End Select

If num <> 0 Then
For i = 0 To (2 ^ num) - 1
Frame7(i).Visible = True
Frame7(i).Caption = X
bb = I
bi = Form8.binary(bb)
For ii = 0 To num - 1
str$ (bi(ii))
Y = str$(bi(ii))
Frame7(i).Caption = Frame7(i).Caption & Y
Next ii
Next i
```

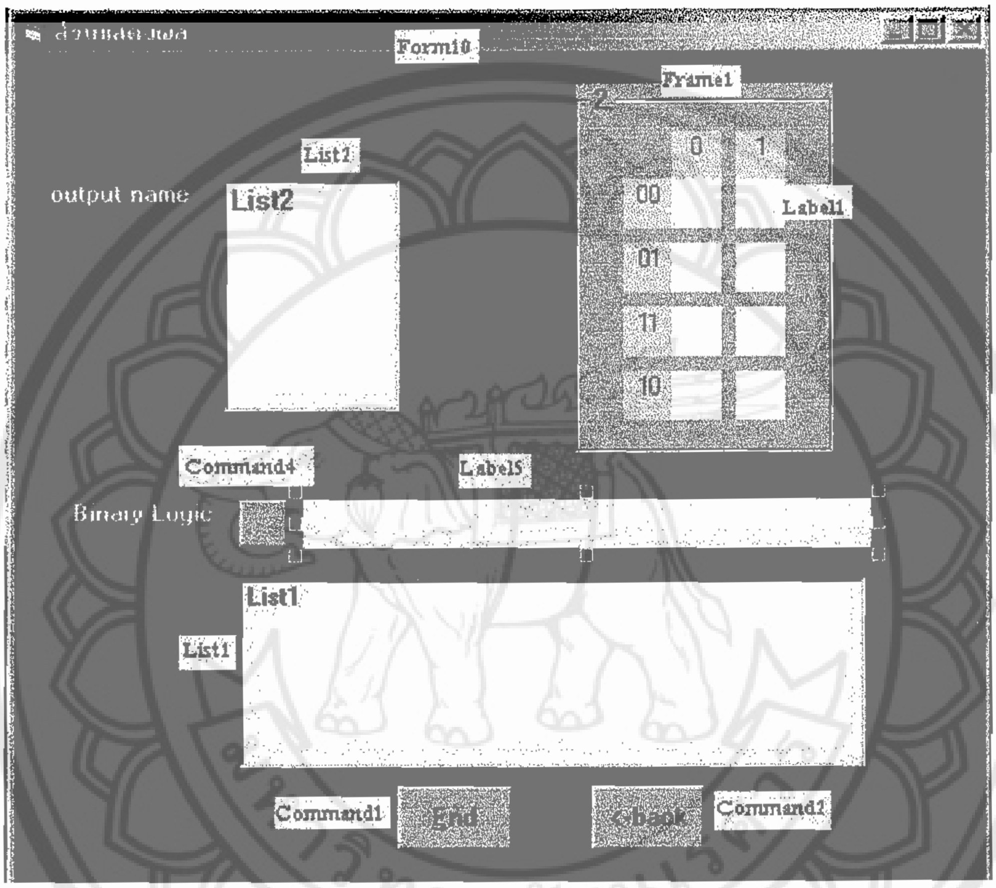
```
Else
Frame7(0).Visible = True
End If
add = 0

For i = 0 To (2 ^ num) - 1
dd = 0
cc = 0
If inf = outf Then
For ii = 0 To inf - 1
Frame8(cc + add).Caption = inff(ii)
Frame8(cc + add).Visible = True
cc = cc + 1
Next ii
Else
For ii = 0 To outf - 1
For bb = 0 To 1
Frame8(cc + add).Caption = Frame8(cc + add).Caption & inff(dd)
Frame8(cc + add).Visible = True
dd = dd + 1
Next bb
cc = cc + 1
Next ii
End If
add = add + 3
Next i

For i = 0 To (2 ^ num) - 1
adde = 0
For ii = 1 To outf
For cc = 0 To sta - 1
bb = ii * 2
PSS = stonum(Label1(cc).Caption, bb)
NSS = stonum(Label2(cc + addt).Caption, bb)

If NSS Like "-" Then
If typeff = 1 Then
Label10(adde + addce + cc).Caption = "dd"
Else
Label10(adde + addce + cc).Caption = "d"
End If
Else
PS = Val(PSS)
NS = Val(NSS)
Label10(adde + addce + cc).Caption = flipflop(typeff, PS, NS)
End If
Next cc
adde = adde + 8
Next ii
addt = addt + 8
addce = addce + 24
Next i
End Sub

Sub putff()
flipflop(0, 0, 0) = "0"
flipflop(0, 0, 1) = "1"
flipflop(0, 1, 0) = "0"
flipflop(0, 1, 1) = "1"
flipflop(1, 0, 0) = "0d"
flipflop(1, 0, 1) = "1d"
flipflop(1, 1, 0) = "d1"
flipflop(1, 1, 1) = "d0"
flipflop(2, 0, 0) = "0"
flipflop(2, 0, 1) = "1"
flipflop(2, 1, 0) = "1"
flipflop(2, 1, 1) = "0"
End Sub

Private Sub Command1_Click()
Form9.Visible = False
Load Form6
Form6.Show
End Sub
Private Sub Command2_Click()
Unload Form8
Load Form8
Form8.Show
Unload Form9
End Sub

Private Sub Command3_Click()
End
End Sub

Private Sub Form_Load()
unk = Form7.cuts(Form7.Text1.Text)
num = Form7.num
output = Form7.cuts(Form7.Text2.Text)
numout = Form7.numout
```

```
State = Form7.cuts(Form7.Text3.Text)
sta = Form7.sta

If Form7.Option1.value = True Then
typeff = 2
End If
If Form7.Option2.value = True Then
typeff = 1
End If
If Form7.Option3.value = True Then
typeff = 0
End If
putff
findff
stassign
trantable
excitable
If num < 3 Then
List1.AddItem "State Assignment"
List1.AddItem "Transition Table"
List1.AddItem "Excitation Table"
Else
List1.AddItem "State Assignment"
List1.AddItem "Transition Table1"
List1.AddItem "Excitation Table1"
List1.AddItem "Transition Table2"
List1.AddItem "Excitation Table2"
End If
List1.ListIndex = 0
End Sub
```

```
Private Sub List1_Click()
Frame4.Visible = False
Frame1(0).Visible = False
Frame5(1).Visible = False
Frame1(1).Visible = False
Frame5(0).Visible = False

Select Case List1.ListIndex
Case 0
Frame4.Visible = True
Case 1
Frame1(0).Visible = True
Case 2
Frame5(1).Visible = True
Case 3
Frame1(1).Visible = True
Case 4
Frame5(0).Visible = True
End Select
End Sub
```

Form10 (ส่วนแสดงผล)

```
Dim choice As Integer
Dim schoice(9) As Integer
Dim lo As Integer
Dim numout As Integer
Dim output() As String
Dim dc(12) As Integer
Dim unk(9) As String
Dim num As Integer
Dim tnumf As Integer
```



```
Dim tnuml As Integer
Dim k(4, 12) As Integer
Dim a(12) As Integer
Dim hj(12) As Integer
Dim hjj(12, 32) As Integer
Dim n(4, I2) As Integer
Dim arrays(12, 32) As Integer
Dim strnum(4, 12, 10, 100, 5) As Integer
Dim cycle(4, 12, 100, 100) As Integer
Dim cy(4, 12, 100) As Integer
Dim cy(4, 12, 100) As Integer
Dim messp(12, 50) As String
Dim logic As String
Dim firstx As Integer
Dim firsty As Integer
Dim xy As Integer
Dim State() As String
Dim sta As Integer
Dim frameindex As Integer
Dim oldlist As Integer
Dim outff(5) As String
Dim outf As Integer
Dim inff(10) As String
Dim inf As Integer

Function checkdo(use() As Integer, u As Integer) As Integer
Dim z As Integer
Dim zz As Integer
Dim zi As Integer
Dim arr(32) As Integer
Dim aa As Integer
Dim yes As Integer
Dim out As Integer
Dim str(30, 5) As Integer
Dim cyc(30, 35) As Integer
Dim cyy(35) As Integer
Dim h(32) As Integer
Dim bb As Integer

yes = 0
```

```
out = 0


For z = 0 To u
For zz = 0 To cy(choice, lo, use(z))
For zi = 0 To a(lo)
If cycle(choice, lo, use(z), zz) = (arrays(lo, zi) + tnumf) Then
arr(zi) = 1
End If
Next zi
Next zz
Next z

If choice = 1 Then

For z = 0 To u
For zz = 0 To cy(choice, lo, use(z))
For zi = 0 To hj(lo)
If cycle(choice, lo, use(z), zz) = (hjj(lo, zi) + tnumf) Then
h(zi) = 1
End If
Next zi
Next zz
Next z

bb = -1

For z = 0 To hj(lo)
If h(z) = 1 Then
bb = bb + 1
```

```
End If
Next z

End If

aa = -1

For z = 0 To a(lo)
If arr(z) = 1 Then
aa = aa + 1
End If
Next z

If choice = 1 Then

If (aa = a(lo)) And (bb = hj(lo)) Then
yes = 1
out = 1
End If

Else
If (aa = a(lo)) Then
yes = 1
out = 1
End If
End If

If yes = 1 Then
```

```
For z = 0 To u
For zz = 0 To num - 1
str(z, zz) = strnum(choice, lo, n(choice, lo), use(z), zz)
Next zz
For zz = 0 To cy(choice, lo, use(z))
cyc(z, zz) = cycle(choice, lo, use(z), zz)
Next zz
cyy(z) = cy(choice, lo, use(z))
Next z

For z = 0 To u
For zz = 0 To num - 1
strnum(choice, lo, n(choice, lo), z, zz) = str(z, zz)
Next zz
For zz = 0 To cyy(z)
cycle(choice, lo, z, zz) = cyc(z, zz)
Next zz
cy(choice, lo, z) = cyy(z)
Next z
k(choice, lo) = u
End If

checkdo = out

End Function

Sub checkdcare()
Dim i As Integer
Dim ii As Integer
Dim X As Integer
Dim numdon As Integer
Dim str(30, 5) As Integer
Dim cyc(30, 35) As Integer
Dim cyy(35) As Integer
Dim u As Integer
Dim use(20) As Integer
Dim z As Integer
Dim zz As Integer
Dim ken As Integer
Dim ryu As Integer
Dim same As Integer
Dim isthis As Integer
Dim c As Integer
Dim sho(20) As Integer
Dim gon(32, 32) As Integer
Dim g(32) As Integer
Dim s As Integer
Dim t As Integer
Dim w As Integer
Dim che As Integer
Dim q As Integer
Dim lee(32) As Integer
Dim r As Integer
Dim every As Integer

t = -1
u = -1

For i = 0 To k(choice, lo)
numdon = -1
```

```
For ii = 0 To cy(choice, lo, i)
For X = 0 To dc(lo)
If cycle(choice, lo, i, ii) = dcare(lo, X) Then
numdon = numdon + 1
End If
Next X
Next ii
If numdon <> cy(choice, lo, i) Then
u = u + 1
use(u) = i
End If
Next i

If u <> k(choice, lo) Then

If u <> -1 Then
For z = 0 To u
For zz = 0 To num - 1
str(z, zz) = strnum(choice, lo, n(choice, lo), use(z), zz)
Next zz
For zz = 0 To cy(choice, lo, use(z))
cyc(z, zz) = cycle(choice, lo, use(z), zz)
Next zz
cyy(z) = cy(choice, lo, use(z))
Next z

For z = 0 To u
For zz = 0 To num - 1
strnum(choice, lo, n(choice, lo), z, zz) = str(z, zz)
Next zz
```

```
For zz = 0 To cyy(z)
cycle(choice, lo, z, zz) = cyc(z, zz)
Next zz
cy(choice, lo, z) = cyy(z)
Next z
k(choice, lo) = u

Else

a(lo) = -1
End If
End If

same = -1
For i = 0 To a(lo)
For ii = 0 To dc(lo)
If arrays(lo, i) = dcare(lo, ii) Then
same = same + 1
End If
Next ii
Next i

For X = 0 To k(choice, lo) - 1
For z = X + 1 To k(choice, lo)
If cy(choice, lo, X) = cy(choice, lo, z) Then
every = 1
Else
every = 0
End If
Next z
Next X

If same <> a(lo) Then

u = -1

For i = 0 To k(choice, lo)
ken = 0
same = 0
isthis = 0
For ii = 0 To cy(choice, lo, i)
For X = 0 To dc(lo)
If cycle(choice, lo, i, ii) = dcare(lo, X) Then
isthis = 1
End If
Next X
Next ii

If isthis = 1 Then

For X = 0 To cy(choice, lo, i)
c = 0
ryu = 0
For Y = 0 To dc(lo)
If cycle(choice, lo, i, X) = dcare(lo, Y) Then
c = 1
End If
Next Y

If c <> 1 Then
```

```
ken = ken + 1
For z = 0 To k(choice, lo)
sho(z) = 0
For zz = 0 To cy(choice, lo, z)
If cycle(choice, lo, i, X) = cycle(choice, lo, z, zz) Then
sho(z) = sho(z) + 1
ryu = ryu + 1
End If
Next zz
Next z

If ryu > 1 Then
same = same + 1
End If
End If

Next X
Else
same = 1
End If



If (same <> ken) Or (every = 1) Then

che = 0
If t <> -1 Then
For q = 0 To t
lee(q) = -1
Next q
For s = 0 To cy(choice, lo, i)
For w = 0 To hj(lo)
If cycle(choice, lo, i, s) = (hjj(lo, w) + tnumf) Then
For q = 0 To t
For r = 0 To g(t)
If gon(q, r) = cycle(choice, lo, i, s) Then
lee(q) = lee(q) + 1
End If
Next r
Next q
End If
Next w
Next s

For s = 0 To t
If lee(s) = g(s) Then
che = 1
End If
Next s
End If

If che = 0 Then
t = t + 1
g(t) = -1
For s = 0 To cy(choice, lo, i)
For w = 0 To hj(lo)
If cycle(choice, lo, i, s) = (hjj(lo, w) + tnumf) Then
g(t) = g(t) + 1
gon(t, g(t)) = cycle(choice, lo, i, s)
End If
Next w
Next s
u = u + 1
use(u) = i
End If

Else
For X = i + 1 To k(choice, lo)
If same = sho(X) Then

che = 0
If t <> -1 Then
For q = 0 To t
lee(q) = -1
Next q
For s = 0 To cy(choice, lo, i)
For w = 0 To hj(lo)
If cycle(choice, lo, i, s) = (hjj(lo, w) + tnumf) Then
For q = 0 To t
For r = 0 To g(t)
If gon(q, r) = cycle(choice, lo, i, s) Then
lee(q) = lee(q) + 1
End If
Next r
Next q
End If
Next w
Next s

For s = 0 To t
If lee(s) = g(s) Then
```

```
che = 1
End If
Next s
End If

If che = 0 Then
t = t + 1
g(t) = -1
For s = 0 To cy(choice, lo, i)
For w = 0 To hj(lo)
If cycle(choice, lo, i, s) = (hjj(lo, w) + tnumf) Then
g(t) = g(t) + 1
gon(t, g(t)) = cycle(choice, lo, i, s)
End If
Next w
Next s
u = u + 1
use(u) = X
End If

End If

Next X

End If
Next i

If u <> k(choice, lo) Then

If u <> -1 Then
```

```
For z = 0 To u
For zz = 0 To num - 1
str(z, zz) = strnum(choice, lo, n(choice, lo), use(z), zz)
Next zz
For zz = 0 To cy(choice, lo, use(z))
cyc(z, zz) = cycle(choice, lo, use(z), zz)
Next zz
cyy(z) = cy(choice, lo, use(z))
Next z

For z = 0 To u
For zz = 0 To num - 1
strnum(choice, lo, n(choice, lo), z, zz) = str(z, zz)
Next zz
For zz = 0 To cyy(z)
cycle(choice, lo, z, zz) = cyc(z, zz)
Next zz
cy(choice, lo, z) = cyy(z)
Next z
k(choice, lo) = u

Else

a(lo) = -1
End If
End If
End If

End Sub
```

```
Sub fcycle()

Dim i As Integer
Dim add As Integer
Dim size As Integer
Dim ii As Integer
Dim match As Integer
Dim X As Integer
Dim xx As Integer
Dim z As Integer
Dim sum(100) As Integer
Dim sumpair As Integer
Dim sumall As Integer
Dim u As Integer
Dim use(100) As Integer
Dim out As Integer
Dim s As Integer
Dim sumpuk As Integer
Dim zz As Integer
Dim zi As Integer
Dim za As Integer
Dim zb As Integer
Dim yes As Integer
Dim compare As Integer
Dim nub As Integer
Dim zg As Integer
Dim doo As Integer
out = 0

If ((a(lo) <> -1) And (a(lo) <> ((2 ^ num) - 1))) Then
```

```
For i = 0 To k(choice, lo)
add = 0
size = -1
For ii = 0 To num - 1
If strnum(choice, lo, n(choice, lo), i, ii) = -1 Then
size = size + 1
cycle(choice, lo, i, size) = 2 ^ (num - 1 - ii)
Else
add = add + (strnum(choice, lo, n(choice, lo), i, ii) * (2 ^ (num - 1 - ii)))
End If
Next ii

s = size

For X = 0 To s - 1
For xx = X + 1 To s
size = size + 1
cycle(choice, lo, i, size) = cycle(choice, lo, i, X) + cycle(choice, lo, i, xx)
Next xx
Next X

For match = 1 To s - 1
For X = 0 To s - 1 - match
For xx = X + 1 To s - match
sumpair = cycle(choice, lo, i, X)
For z = xx To xx + match - 1
sumpair = sumpair + cycle(choice, lo, i, z)
Next z
```

```
For zz = z To s
size = size + 1
cycle(choice, lo, i, size) = sumpair + cycle(choice, lo, i, zz)
Next zz
Next xx
Next X
Next match

size = size + 1
cycle(choice, lo, i, size) = 0

For X = 0 To size
cycle(choice, lo, i, X) = cycle(choice, lo, i, X) + add + tnumf
Next X

cy(choice, lo, i) = size
Next i

If (k(choice, lo) <> 0) And (k(choice, lo) <> 1) Then

For i = 0 To k(choice, lo)
sum(i) = 0

For ii = 0 To cy(choice, lo, i)
sum(i) = sum(i) + cycle(choice, lo, i, ii)
Next ii

Next i

sumall = 0
```

```
For i = 0 To a(lo)
sumall = sumall + arrays(lo, i) + tnumf
Next i


For i = 0 To k(choice, lo) - 1
sumpair = 0
u = -1
u = u + 1
use(u) = i
sumpair = sumpair + sum(i)
For ii = i + 1 To k(choice, lo)
u = u + 1
use(u) = ii
If ((sumpair + sum(ii)) = sumall) Then
out = checkdo(use(), u)
End If
If out = 1 Then
Exit For
End lf
u = u - 1
Next ii
If out = 1 Then
Exit For
End lf
Next i

If out = 0 Then
For match = 1 To k(choice, lo) - 1
For i = 0 To k(choice, lo) - 1 - match
For xx = i + 1 To k(choice, lo) - match + 1
u = -1
u = u + 1
use(u) = i
sumpair = sum(i)
For z = xx To xx + match - 1

u = u + 1
use(u) = z
sumpair = sumpair + sum(z)
Next z

For ii = z To k(choice, lo)
u = u + 1
use(u) = ii

For zz = 0 To u - 1
For zi = zz + 1 To u
If cy(choice, lo, zz) > cy(choice, lo, zi) Then
compare = cy(choice, lo, zi) + 1
Else
compare = cy(choice, lo, zz) + 1
End If
nub = 0
For za = 0 To cy(choice, lo, zz)
For zb = 0 To cy(choice, lo, zi)
If cycle(choice, lo, zz, za) = cycle(choice, lo, zi, zb) Then
nub = nub + 1
End If
```

```
Next zb
Next za
If (nub > 0) And (nub < compare) Then
yes = 1
End If
Next zi
Next zz


If ((sumpair + sum(ii) = sumall) Or (yes = 1)) Then

If u <> k(choice, lo) Then
out = checkdo(use(), u)
End If
End If
u = u - 1
yes = 0

If out = 1 Then
Exit For
End If
Next ii

If out = 0 Then
For ii = i + 1 To xx - 1
u = u + 1
use(u) = ii

For zz = 0 To u - 1
For zi = zz + 1 To u
```

```
If cy(choice, lo, zz) > cy(choice, lo, zi) Then
compare = cy(choice, lo, zi) + 1
Else
compare = cy(choice, lo, zz) + 1
End If
nub = 0
For za = 0 To cy(choice, lo, zz)
For zb = 0 To cy(choice, lo, zi)
If cycle(choice, lo, zz, za) = cycle(choice, lo, zi, zb) Then
nub = nub + 1
End If
Next zb
Next za
If (nub > 0) And (nub < compare) Then
yes = 1
End If
Next zi
Next zz


If ((sumpair + sum(ii) = sumall) Or (yes = 1)) Then
doo = 1
For zg = 0 To u - 1
For zz = zg + 1 To u
If use(zg) = use(zz) Then
doo = 0
End If
Next zz
Next zg
If (u <> k(choice, lo)) And (doo = 1) Then
```

```
out = checkdo(use(), u)
End If
End If
u = u - 1
yes = 0

If out = 1 Then
Exit For
End If
Next ii
End If

If out = 1 Then
Exit For
End If
Next xx

If out = 1 Then
Exit For
End If
Next i

If out = 1 Then
Exit For
Else
sumpair = 0
u = -1
u = u + 1
use(u) = k(choice, lo)
sumpair = sumpair + sum(k(choice, lo))
```

```
For xx = 0 To match - 1
u = u + 1
use(u) = xx
sumpair = sumpair + sum(xx)
Next xx
For ii = 0 To match
u = u + 1
use(u) = ii

For zz = 0 To u - 1
For zi = zz + 1 To u
If cy(choice, lo, zz) > cy(choice, lo, zi) Then
compare = cy(choice, lo, zi) + 1
Else
compare = cy(choice, lo, zz) + 1
End If
nub = 0
For za = 0 To cy(choice, lo, zz)
For zb = 0 To cy(choice, lo, zi)
If cycle(choice, lo, zz, za) = cycle(choice, lo, zi, zb) Then
nub = nub + 1
End If
Next zb
Next za
If (nub > 0) And (nub < compare) Then
yes = 1
End If
Next zi
Next zz
```

```
If ((sumpair + sum(ii) = sumall) Or (yes = 1)) Then

If u <> k(choice, lo) Then
out = checkdo(use(), u)
End If
End If
u = u - 1
yes = 0

If out = 1 Then
Exit For
End If
Next ii
End If


Next match
End If

End If

If (dc(lo) > -1) And (choice = 1) Then
checkdcare
End If

za = -1

For z = 0 To k(choice, lo)
For zz = 0 To cy(choice, lo, z)
```

```
za = za + I
cycle(choice, lo, k(choice, lo) + 1, za) = cycle(choice, lo, z, zz)
Next zz
Next z

cy(choice, lo, k(choice, lo) + 1) = za

End If


End Sub

Sub minform()

Dim i As Integer
Dim io As Integer
Dim ii As Integer
Dim relize As Integer
Dim dif As Integer
Dim same As Integer
Dim doo As Integer
Dim rou As Integer
Dim locate As Integer
Dim use(100) As Integer
Dim str() As Integer
Dim c As Integer
Dim stt(5) As Integer
Dim ar As Integer

n(choice, lo) = 0
```

```
k(choice, lo) = -1
dif = 0
same = 0

If (a(lo) = -1) Or (a(lo) = (2 ^ num) - 1) Then
relize = 0
Else
If a(lo) = 0 Then
relize = 0
k(choice, lo) = k(choice, lo) + 1
ar = arrays(lo, 0)
str = binary(ar)
For ii = 0 To num - 1
strnum(choice, lo, n(choice, lo), k(choice, lo), ii) = str(ii)
Next ii
Else
relize = 1
End If
End If


If relize = 1 Then
For i = 0 To a(lo)
k(choice, lo) = k(choice, lo) + 1
ar = arrays(lo, i)
str = binary(ar)
For ii = 0 To num - 1
strnum(choice, lo, n(choice, lo), k(choice, lo), ii) = str(ii)
Next ii
Next i
```

```
End If

Do While relize = 1

relize = 0

For io = 0 To k(choice, lo) - 1
For i = io + 1 To k(choice, lo)
dif = 0
For ii = 0 To num - 1
If strnum(choice, lo, n(choice, lo), io, ii) <> strnum(choice, lo, n(choice, lo), i, ii) Then
dif = dif + 1
End If
Next ii
If dif = 1 Then
relize = 1
Exit For
Else
relize = 0
End If
Next i
If relize = 1 Then
Exit For
End If
Next io

If relize = 1 Then
rou = k(choice, lo)
k(choice, lo) = -1
doo = 1
```

```
For i = 0 To rou
use(i) = 0
Next i

For io = 0 To rou - 1
For i = io + 1 To rou
dif = 0
For ii = 0 To num - 1
If strnum(choice, lo, n(choice, lo), io, ii) <> strnum(choice, lo, n(choice, lo), i, ii) Then
dif = dif + 1
locate = ii
End If
Next ii

If dif = 1 Then
For ii = 0 To num - 1
stt(ii) = strnum(choice, lo, n(choice, lo), i, ii)
Next ii
stt(locate) = -1

For c = 0 To k(choice, lo)
same = 0
For ii = 0 To num - 1
If strnum(choice, lo, n(choice, lo) + 1, c, ii) = stt(ii) Then
same = same + 1
End If
Next ii
If same = num Then
doo = 0
```

```
Exit For
Else
doo = 1
End If
Next c

If doo = 1 Then
k(choice, lo) = k(choice, lo) + 1
For ii = 0 To num - 1
strnum(choice, lo, n(choice, lo) + 1, k(choice, lo), ii) = strnum(choice, lo, n(choice, lo), i, ii)
Next ii
strnum(choice, lo, n(choice, lo) + 1, k(choice, lo), locate) = -1
End If
use(i) = 1
use(io) = 1
End If

If dif = 0 Then
k(choice, lo) = k(choice, lo) + 1
For ii = 0 To num - 1
strnum(choice, lo, n(choice, lo) + 1, k(choice, lo), ii) = strnum(choice, lo, n(choice, lo), i, ii)
Next ii
use(i) = 1
use(io) = 1
End If

Next i
Next io

For i = 0 To rou
```

```
If use(i) = 0 Then
k(choice, lo) = k(choice, lo) + 1
For ii = 0 To num - 1
strnum(choice, lo, n(choice, lo) + 1, k(choice, lo), ii) = strnum(choice, lo, n(choice, lo), i, ii)
Next ii
End If

Next i

n(choice, lo) = n(choice, lo) + 1
End If

Loop

End Sub


Function binary(z As Integer) As Integer()
Dim gg As Integer
Dim kg As Integer
Dim strr() As Integer
ReDim strr(num)

gg = 2 ^ (num - 1)

Do While Not ((gg = 0) And (z = 0))
z = z - gg
If z >= 0 Then
strr(kg) = 1
kg = kg + 1
```

```
Else
strr(kg) = 0
kg = kg + 1
z = z + gg
End If
gg = gg / 2
Loop

binary = strr

End Function



Sub appear()
Select Case num
Case 2
Frame1(0).Visible = True
Frame1(0).Caption = unk(1) & " \ " & unk(0)
tnumf = 0
tnuml = 3
Case 3
Frame1(1).Visible = True
Frame1(1).Caption = unk(1) & " " & unk(2) & " \ " & unk(0)
tnumf = 4
tnuml = 11
End Select
End Sub

Sub fmess()
Dim mess As String
Dim i As Integer
Dim ii As Integer
Dim aorhj As Integer

If schoice(lo) = 0 Then
aorhj = hj(lo)
Else
aorhj = a(lo)
End If

For i = 0 To 15
messp(lo, i) = ""
Next i

Select Case aorhj
Case -1
If logic = "0" Then
messp(lo, 0) = "1"
Else
messp(lo, 0) = "0"
End If
k(schoice(lo), lo) = k(schoice(lo), lo) + 1
Case ((2 ^ num) - 1)
If dc(lo) <> (2 ^ num) - 1 Then
If logic = "0" Then
messp(lo, 0) = "0"
Else
messp(lo, 0) = "1"
End If
Else
```

```
If logic = "0" Then
messp(lo, 0) = "1"
Else
messp(lo, 0) = "0"
End If
k(schoice(lo), lo) = 0
End If
k(schoice(lo), lo) = k(schoice(lo), lo) + 1
Case Else
Select Case logic
Case "0"
For i = 0 To k(schoice(lo), lo)
For ii = 0 To num - 1

Select Case strnum(schoice(lo), lo, n(schoice(lo), lo), i, ii)
Case 0
If messp(lo, i) = "" Then
mess = mess & "("
Else
mess = mess & "+"
messp(lo, i) = messp(lo, i) & "+"
End If

mess = mess & unk(ii)
messp(lo, i) = messp(lo, i) & unk(ii)

Case 1
If messp(lo, i) = "" Then
mess = mess & "("
Else
mess = mess & "+"
messp(lo, i) = messp(lo, i) & "+"
End If
mess = mess & unk(ii) & "*"
messp(lo, i) = messp(lo, i) & unk(ii) & "*"

End Select

Next ii
mess = mess & ")"
Next i

Case "1"
For i = 0 To k(schoice(lo), lo)
For ii = 0 To num - 1

Select Case strnum(schoice(lo), lo, n(schoice(lo), lo), i, ii)
Case 0
mess = mess & unk(ii) & "*"
messp(lo, i) = messp(lo, i) & unk(ii) & "*"
Case 1
mess = mess & unk(ii)
messp(lo, i) = messp(lo, i) & unk(ii)
End Select

Next ii
If i <> k(schoice(lo), lo) Then
mess = mess & "+"
End If
Next i
```

```
End Select
messp(lo, k(schoice(lo), lo) + 1) = mess
End Select

End Sub

Sub initail()
Dim i As Integer
Dim j As String
Dim lis As Integer
Dim ii As Integer
Dim aorhj As Integer

lis = List2.ListIndex
If schoice(lis) = 0 Then
aorhj = hj(lis)
Else
aorhj = a(lis)
End If


For i = 0 To aorhj
Label1(arrays(lis, i) + tnumf).Caption = logic
Next i

For i = 0 To dc(lis)
Label1(dcare(lis, i)).Caption = "d"
Next i
```

```
If k(schoice(lis), lis) = 0 Then
Label5.Caption = messp(lis, 0)
If aorhj = ((2 ^ num) - 1) Then
For i = tnumf To tnuml
Label1(i).BackColor = QBColor(10)
Next i
End If

If (aorhj <> -1) And (aorhj <> (2 ^ num) - 1) Then
For i = 0 To cy(schoice(lis), lis, k(schoice(lis), lis))
Label1(cycle(schoice(lis), lis, k(schoice(lis), lis), i)).BackColor = QBColor(10)
Next i
End If

Else

If (aorhj <> -1) And (aorhj <> (2 ^ num) - 1) Then
For i = 0 To k(schoice(lis), lis)
List1.AddItem messp(lis, i)
Next i
j = messp(lis, k(schoice(lis), lis) + 1) & " (ALL)"
List1.AddItem j
Label5.Caption = messp(lis, k(schoice(lis), lis) + 1)
For i = 0 To cy(schoice(lis), lis, k(schoice(lis), lis) + 1)
Label1(cycle(schoice(lis), lis, k(schoice(lis), lis) + 1, i)).BackColor = QBColor(10)
Next i

Else
Label5.Caption = messp(lis, 0)
```

```
End If

End If

If k(schoice(lis), lis) > 0 Then
Command4.Visible = True
End If
If dc(lis) = (2 ^ num) - 1 Then
Command4.Visible = False
End If
If Len(Label5.Caption) > 40 Then
Label5.Width = 7000
Label5.Font.size = 8
List1.Width = 7000
End If

End Sub


Sub findff()
Dim i As Integer
Dim X As String
Dim b As Integer

For i = 1 To 3
If (2 ^ i) >= sta Then
outf = i
Exit For
End If
Next i
```

```
For i = 0 To outf - 1
outff(i) = "Q" & str$(outf - 1 - i)
Next i

Select Case Form9.typeff
Case 0
X = "D"
Case 1
X = "JK"
Case 2
X = "T"
End Select

If X = "JK" Then
For i = 0 To outf - 1
inff(b) = "J" & str$(outf - 1 - i)
b = b + 1
inff(b) = "K" & str$(outf - 1 - i)
b = b + 1
Next i
inf = b

Else
For i = 0 To outf - 1
inff(i) = X & str$(outf - 1 - i)
Next i
inf = outf
End If
End Sub

Private Sub Command1_Click()
End
End Sub

Private Sub Command2_Click()
Form6.Visible = True
Unload Form10
End Sub

Private Sub Command4_Click()
If List1.Visible = True Then
List1.Visible = False
Else
List1.Visible = True
End If
End Sub

Private Sub Form_DragDrop(Source As Control, X As Single, Y As Single)
Frame1(frameindex).Top = Y - firsty
Frame1(frameindex).Left = X - firstx
xy = 0
End Sub

Private Sub Form_Load()
Dim i As Integer
Dim ii As Integer
Dim X As Integer
Dim numm As Integer
Dim ss(15) As String
Dim nummo As Integer
Dim Y As Integer
Dim z As Integer
Dim g As Integer
Dim za As Integer
Dim zz As Integer
Dim j As Integer
Dim char As String
Dim vk As Integer
Dim h As Integer

If Form7.Combo1.Text = "Minterm" Then
logic = "1"
Else
logic = "0"
End If

State = Form7.cuts(Form7.Text3.Text)
sta = Form7.sta
output = Form7.cuts(Form7.Text2.Text)
numout = Form7.numout
findff

For i = 0 To outf - 1
unk(i) = outff(i)
Next i
num = outf
```

```
appear

a(g) = -1
dc(g) = -1
For ii = 0 To numout - 1
vk = 0
For i = 0 To (2 ^ outf) - 1
char = Form8.Label5(i + h).Caption

Select Case char
Case logic
a(g) = a(g) + 1
arrays(g, a(g)) = vk

Case "-"
dc(g) = dc(g) + 1
dcare(g, dc(g)) = vk + tnumf

End Select
vk = vk + 1

Next i

g = g + 1
a(g) = -1
dc(g) = -1
h = h + 8
Next ii
```

```
If num > 1 Then

choice = 0
For lo = 0 To numout - 1
minform
fcycle
Next lo

choice = 1

For i = 0 To numout - 1
For ii = 0 To a(i)
hjj(i, ii) = arrays(i, ii)
Next ii
hj(i) = a(i)
Next i

For i = 0 To numout - 1
X = a(i)
For ii = 0 To dc(i)
X = X + 1
arrays(i, X) = dcare(i, ii) - tnumf
Next ii
a(i) = X
Next i
```

```
For lo = 0 To numout - 1
minform
fcycle
Next lo

For i = 0 To numout - 1
If k(0, i) < k(1, i) Then
schoice(i) = 0

For ii = 0 To k(0, i)
kyoko:
For X = 0 To k(1, i)
If cy(0, i, ii) <> cy(1, i, X) Then
For Y = 0 To cy(0, i, ii)
For z = 0 To cy(1, i, X)
If cycle(0, i, ii, Y) = cycle(1, i, X, z) Then

For g = 0 To cy(1, i, X)
cycle(0, i, ii, g) = cycle(1, i, X, g)
Next g
cy(0, i, ii) = cy(1, i, X)

za = -1

For j = 0 To k(0, i)
For zz = 0 To cy(0, i, j)
za = za + 1
cycle(0, i, k(0, i) + 1, za) = cycle(0, i, j, zz)
Next zz
Next j
```

```
cy(0, i, k(0, i) + 1) = za

For za = 0 To num - 1
strnum(0, i, n(0, i), ii, za) = strnum(1, i, n(1, i), X, za)
Next za

If ii = k(0, i) Then
GoTo godai:
Else
GoTo kyoko:
ii = ii + 1
End If
End If
Next z
Next Y
End If
Next X
Next ii
godai:
Else
schoice(i) = 1
End If
Next i

For lo = 0 To numout - 1
fmess
Next lo

End If
```

```
For i = 0 To numout - 1
List2.AddItem output(i)
Next i

List2.ListIndex = 0
oldlist = 0

End Sub



Private Sub Frame1_DragOver(Index As Integer, Source As Control, X As Single, Y As Single,
State As Integer)

If xy = 0 Then
frameindex = Index
firstx = X
firsty = Y
xy = 1
End If
End Sub

Private Sub List1_Click()
Dim i As Integer
Dim lis As Integer

lis = List2.ListIndex

Label5.Caption = messp(lis, List1.ListIndex)

For i = tnumf To tnuml
```

```
Label1(i).BackColor = QBColor(15)
Next i

For i = 0 To cy(schoice(lis), lis, List1.ListIndex)
Label1(cycle(schoice(lis), lis, List1.ListIndex, i)).BackColor = QBColor(10)
Next i

List1.Visible = False
End Sub

Private Sub List2_Click()
Dim i As Integer
Dim h As String
Dim ii As Integer
Dim same As Integer

If num = 1 Then

For ii = 0 To 1
h = " " & Form8.Label5(ii).Caption
If h = str$(ii) Then
same = same + 1
Else
If Form8.Label5(ii).Caption = "-" Then
hasd = hasd + 1
End If
End If
Next ii

If hasd = 0 Then
```

```
Select Case same
Case 0
Label5.Caption = "Q0*"
Case 1
Label5.Caption = Form8.Label5(0).Caption
Case 2
Label5.Caption = "Q0"
End Select
Else
Select Case same
Case 0
Label5.Caption = "Q0*"
Case 1
Label5.Caption = "Q0"
End Select
End If
Else
For i = tnumf To tnuml
Label1(i).Caption = ""
Label1(i).BackColor = QBColor(15)
Next i
If List1.ListCount > 0 Then
For i = 0 To k(schoice(oldlist), oldlist) + 1
List1.RemoveItem k(schoice(oldlist), oldlist) + 1 - i
Next i
End If
oldlist = List2.ListIndex
initail
End If
End Sub
```