المحاضرة التاسعة : المعالج processer

عند كتابة برنامج بأحد اللغات العالية المستوى يقوم برنامج الترجمة compiler بتحويل البرنامج الى برنامج هدفي بالغة الألة..... وبعدها يتم تحميل هذا البرنامج الهدفي في الذاكرة الرئيسية....

الأن يقوم المعالج بتنفيذ هذا البرنامج على مرحلتين:

- المرحلة الأولى مرحلة التجهيز FETCH.
- المرحلة الثانية مرحلة التنفيذ EXECUTE.

هناك الكثير من أنواع المعالج المتوفرة ، سنقوم بدراسة نوع صغير جدا فقط للتوضيح.

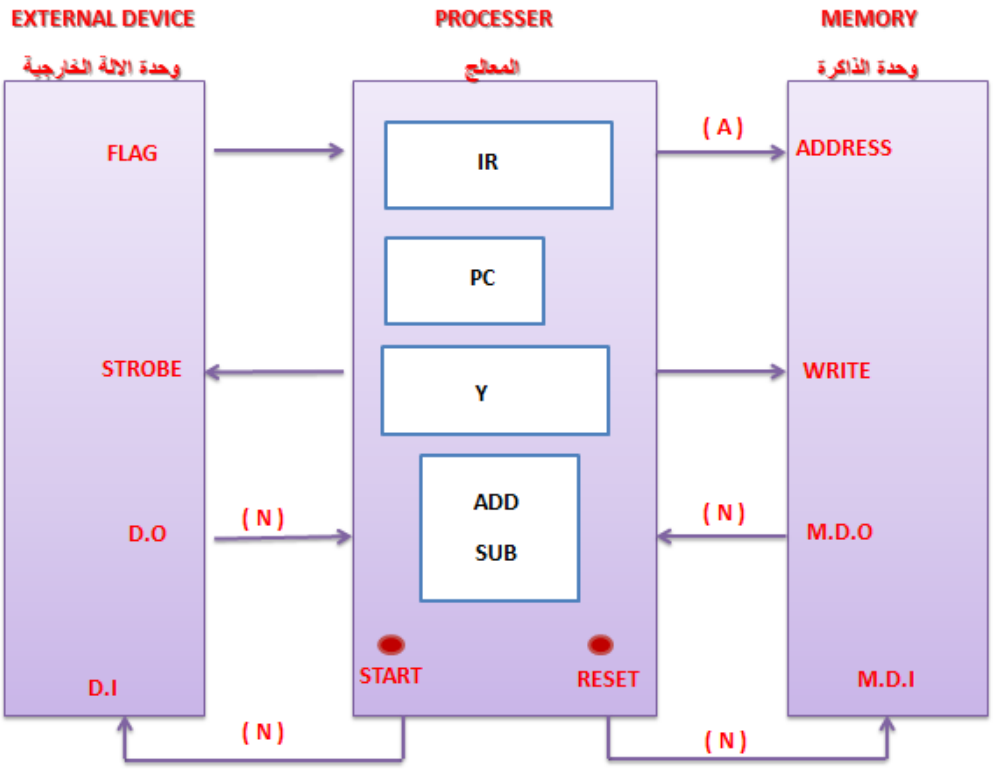
يحتوي المعالج التوضيحي على مسجل التعليمة Instruction Register **(IR)** لتخزين نسخة من التعليمة المراد تنفيذها من برنامجنا الموجود في الذاكرة الرئيسية المراد تنفيذه.

ويحتوي معالجنا التوضيحي على مسجل العداد Program Counter **(PC)** لتخزين عنوان التعليمة التالية للتنفيذ في برنامجنا المراد تنفيذه في الذاكرة الرئيسية.

أيضا ، يحتوي معالجنا التوضيحي على مسجل ذو أغراض عامة مسجل **Y**.

أيضا يحتوي معالجنا التوضيحي ألة للجمع ADD وألة للطرح SUB.

كذلك يحتوي معالجنا التوضيحي Start Button و Reset Button .



تشغيل المعالج:

عند تشغيل المعالج التوضيحي تكون محتويات مسجل العداد PC=0.

المرحلة الأولى مرحلة FETCH :مرحلة التجهيز تتكون من خطوتين:

الخطوة الأولى:

يتم أخذ نسخة من أول تعليمة من البرنامج المراد تنفيذه في الذاكرة الرئيسية السطر صفر وتخزينها في مسجل IR بالمعالج التوضيحي

IR <------ M(0)

الخطوة الثانية:

تتغير محتويات مسجل PC لتشير الى عنوان التعليمة التالية للتنفيذ في برنامجنا المراد تنفيذه

PC <----- PC + 1

المرحلة الثانية مرحلة EXECUTE

فيها يتم ترجمة وتفسير وتنفيذ تعليمة برنامجنا الموجود نسخة منها داخل مسجل IR.

الأن يقوم معالجنا التوضيحي للرجوع الى مرحلة FETCH من جديد لتجهير التعليمة التالية للتنفيذ في برنامجنا المراد تنفيذه حيث يشير إليها قيمة مسجل PC

IR <------- M(PC)

PC <------ PC +1

والأن يقوم معالجنا التوضيحي للرجوع الى مرحلة EXECUTE حيث يتم ترجمة وتفسير وتنفيذ تعليمة برنامجنا الموجود نسخة منها داخل مسجل IR

....
....
وهكذا بين مرحلة FETCH
ومرحة EXECUTE
....
....
الى أين تأتي تعليمة من برنامجنا المراد تنفيذه في الذاكرة الرئيسية يكون ترجمتها وتفسيرها النهاية فيتوقف المعالج.

في مرحلة التنفيذ EXECUTE يتم تقسيم محتويات IR الى جزئيين:

الجزء الأيسر : يتكون من 3 خانات (bits) اليسرى ويسمى

OPERATION CODE (OP)

ومن خلاله يتم معرفة نوع الأمر المراد تنفيذه وذلك بالرجوع الى جدول التعليمات الخاص للمعالج التوضيحي .

الجزء الايمن : وهو باقي الخانات ويسمى الحقل Z

جدول التعليمات INSTRUCTIONS SET الخاص للمعالج التوضيحي

op	INSTRUCTIONS	EXECUATION
000	LOADI	Y<----- Z
001	STORE	M(Z)<---- Y
010	JUMP	PC <----- Z
011	LOAD	Y <----- M(Z)
100	ADD	Y <---- Y+M(Z)
101	SUB	Y <----- Y- M(Z)
110	JUMPFL	if FLAG=1 PC<---- Z
111 (d=0001)	DATAIN	Y<----- EX.DV
111(d=0010)	SKIPNG	if Y neg. PC<---- PC+1
111 (d=0100)	STROBE	EX.DV. <--- Y
111 (d=1000)	HULT	STOP

مثال: قم بمتابعة هذا البرنامج

0 01100000111
1 10000001000
2 01000000101
3 10100001111
4 11000001010
5 00000010000
6 11100001000
7 00000001100
8 00000000110

الحل:

عند تشغيل المعالج التوضيحي تكون محتويات المسجل

PC =0

الأن عملية FETCH

1- IR <----- M(PC)
IR= 01100000111

2. PC <----- PC +1
PC= 1

الأن عملية EXECUTE

تقسم محتويات IR الى

OP code[011]
Z [00000111]

بالرجوع الى جدول التعليمات الخاص للمعالج التوضيحي نجد [011]OP تعني

عملية LOAD

Y <----- M(Z)
Y <------M(7)
Y=[00000001100]
Y = 12

الأن عملية FETCH

1- IR <----- M(PC)
IR <------ M(1)
IR <------ 10000001000

2- PC <----- PC +1
PC <----- 2

الأن عملية EXECUTE : تقسم محتويات IR الى

OP code[100]
Z [00001000]

وبالرجوع الى جدول التعليمات الخاص للمعالج التوضيحي نجدد [100]OP تعني

عملية ADD

Y<----- Y+M(Z)
Y <------ 12 + M(8)
Y <------ 12 + 6

الأن عملية FETCH

1- IR<------ M(PC)
IR <------ M(2)
IR=01000000101

2- PC <------ PC +1
PC=3

الأن عملية EXECUTE : تقسم محتويات IR الى

OP code [010]
Z [00000101]

وبالرجوع الى جدول التعليمات الخاص للمعالج التوضيحي نجد [010]OP تعني عملية JUMP

PC <----- Z
PC = 5

الأن عملية FETCH

1-IR <------ M(PC)
IR <------ M(5)
IR <------00000010000

2-PC<----- PC +1
PC =6

الأن عملية EXECUTE : تقسم محتويات IR الى

OP code [000]
Z[00010000]

وبالرجوع الى جدول التعليمات الخاص للمعالج التوضيحي نجد [000] OP تعني

عملية LOADI

Y <----- Z
Y =[00010000]
Y = 16

الأن عملية FETCH

1- IR <----- M(PC)
IR <------ M(6)
IR =[11100001000]

2- PC <------ PC +1
PC = 7

الأن عملية EXECUTE : تقسم محتويات IR الى

OP code [111]

في هذه الحالة نحتاج الحقل d وهو 4خانات اليمنى من Z وهي [1000]

Z[00001000]

وبالرجوع الى جدول التعليمات الخاص للمعالج التوضيحي نجد [111]OP والحقل [1000]d تعني

عملية STOP

فيقف المعالج عن التشغيل

مثال : قم بمتابعة هذا البرنامج

0	00000001010
1	00100000100
2	11100000100
3	11000000111
4	11111111000
5	10000000111
6	00000001111
7	10100000110
8	11100000010
9	01000000011
10	11100000001
11	11100001000
12	01100000110

الحل:

عند تشغيل المعالج التوضيحي تكون محتويات

المسجل PC=0

عملية FETCH

1- IR <------ M(0)
IR=[00000001010]

2- PC <----- PC +1
PC=1

عملية EXECUTE :

OP CODE [000]
Z [00001010]

Y <----- Z
Y=[00001010]
Y= 10

عملية FETCH

1- IR <----- M(1)
IR =[00100000100]

2- PC=2

عملية EXECUTE :

OP CODE [001]
Z=[00000100]

M(Z) <---- Y
M(4) = 10
M(4)=[00000001010]

عملية FETCH

1- IR <---- M(2)
IR =[11100000100]

2- PC =3

عملية EXECUTE :

OP CODE [111]
نحتاج الحقل d
d[0100]
نوع العملية STROBE

EX.DV. <------ Y
هنا تصبح قيمة FLAG=1
EX.DV. = 10

عملية FETCH

1- IR <---- M(3)
IR =[11000000111]

2- PC =4

عملية EXECUTE :

OP CODE [110]
Z [00000111]
التعليمة JUMPFL
if FLAG =1

إذا تحقق الشرط نعم

PC <---- Z
PC =7

عملية FETCH

1- IR <----- M(7)
IR =[10100000110]

2- PC = 8

عملية EXECUTE :

OP CODE [101]
Z =[00000110]

Y <------ Y - M(Z)
Y <----- 10 - M(6)
Y <------ 10 - 15
Y = -5

عملية FETCH

1- IR <---- M(8)
IR =[11100000010]

2- PC =9

عملية EXECUTE :

OP CODE [111]
d[0010]
عملية SKIPNG
if Y سالبة نعم تحقق الشرط ، إذا

PC<-----PC +1
PC=10

عملية FETCH

1- IR <----- M(10)
IR =[11100000001]

2- PC =11

عملية EXECUTE :

OP CODE [111]
d[0001] من الجدول نجد انها عملية DATAIN

Y <------ EX.DV
Y =10 / FLAG=0 هنا تصبح قيمة

عملية FETCH

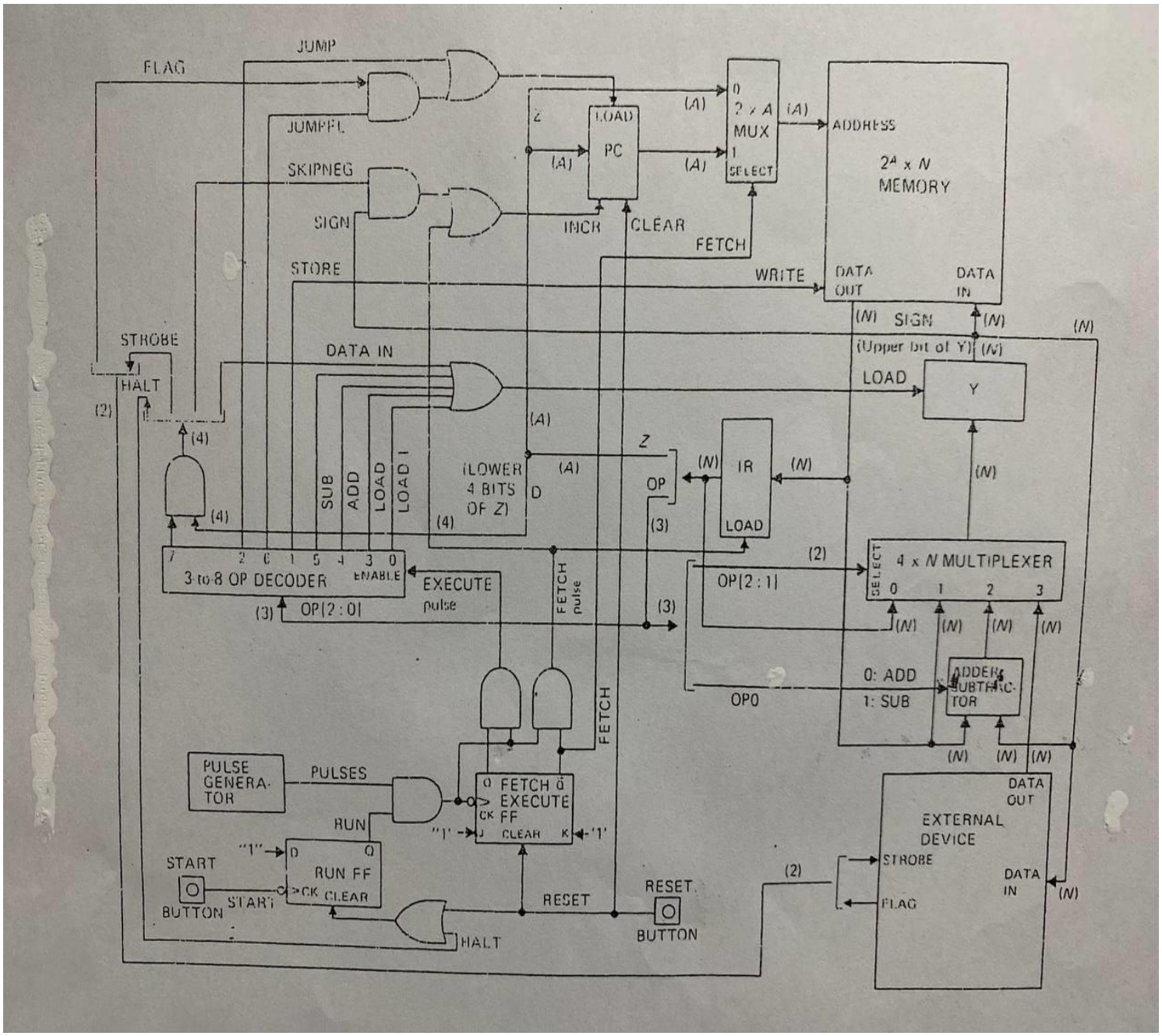
1- IR <----- M(11)
IR =[11100001000]

2- PC = 12

عملية EXECUTE

OP CODE [111]
d[1000]

STOP


JUMP
FLAG
JUMPFL
SKIPNEG
SIGN
STORE
STROBE
HALT
DATA IN
SUB
ADD
LOAD
LOAD I
3 to 8 OP DECODER
ENABLE
EXECUTE pulse
FETCH pulse
OP[2:0]
LOWER 4 BITS OF Z
LOAD
PC
INCR
CLEAR
FETCH
2 x A MUX
SELECT
ADDRESS
2^A x N MEMORY
WRITE
DATA OUT
DATA IN
SIGN (Upper bit of Y)
Y
IR
OP
OP[2:1]
4 x N MULTIPLEXER
0: ADD
1: SUB
OP0
ADDER/SUBTRACTOR
PULSE GENERATOR
PULSES
RUN
FETCH & EXECUTE FF
CK
J
CLEAR
K
RUN FF
START
START BUTTON
RESET
RESET BUTTON
EXTERNAL DEVICE
STROBE
FLAG
DATA OUT
DATA IN