

DATE : Lec 1

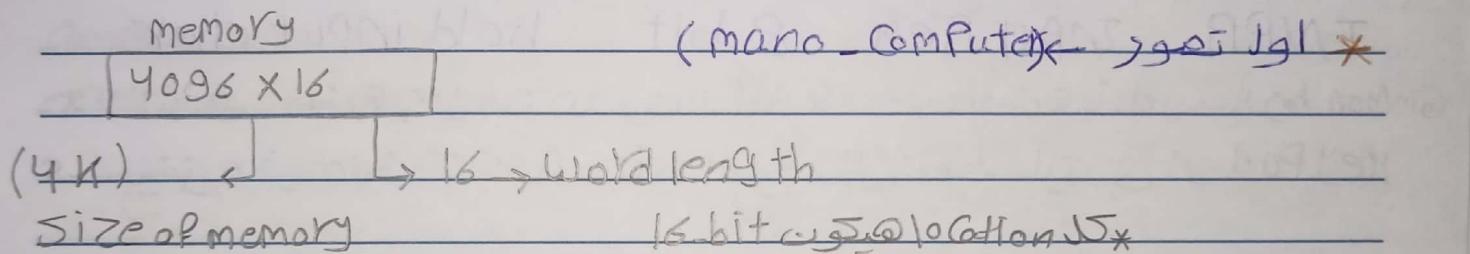
SUBJECT :

Basic Computer organization and Design

* The organization of the computer is defined by :-

- 1- Its internal register
- 2- Timing and control structure
- 3- Set of the instruction

* Organization is a sequence of micro-operations it performs on data stored in its internal registers



* Internal register

DR	Data register	16-bits	hold the operand (Read/Write)
AR	Address register	12-bits	hold memory Address
COMMON BUS	common bus	12 bits	دبوس تواقة وسائل علامة تواقة
MSN	MSN (MSB)	11	العنوان المقامرة (Common Bus) أو باختصار

PC	Program Counter	12-bits	hold the next instruction Address.
----	-----------------	---------	------------------------------------

DATE :

AC Accumulator 16-bits Process Register

$AC \leftarrow M[AB]$ (LDA)
 $M[AB] \leftarrow AC$ (STA)

IA Instruction Reg 16-bit hold current instruction

TR Temporary Reg 16-bit hold temporary data
وهو寄存器 لـ temporary data ويعتبر Operands

INPR Input Reg 8-bit hold input char

١- الـ 8-bit من ذيروه حتى اربعينيـة زد الى اربعـينيـة ١٢-bit

OUT R output Reg 8-bit hold output char

البيانات ترسل على حافلة Common bus لـ 8086

specific output that is selected for the bus lines at any given time is determined from the selection variables S_2 , S_1 , and S_0 . The number along each output shows the decimal equivalent of the required binary selection. For example, the number along the output of DR is 3. The 16-bit outputs of DR are placed on the bus when $S_2S_1S_0 = 011$.

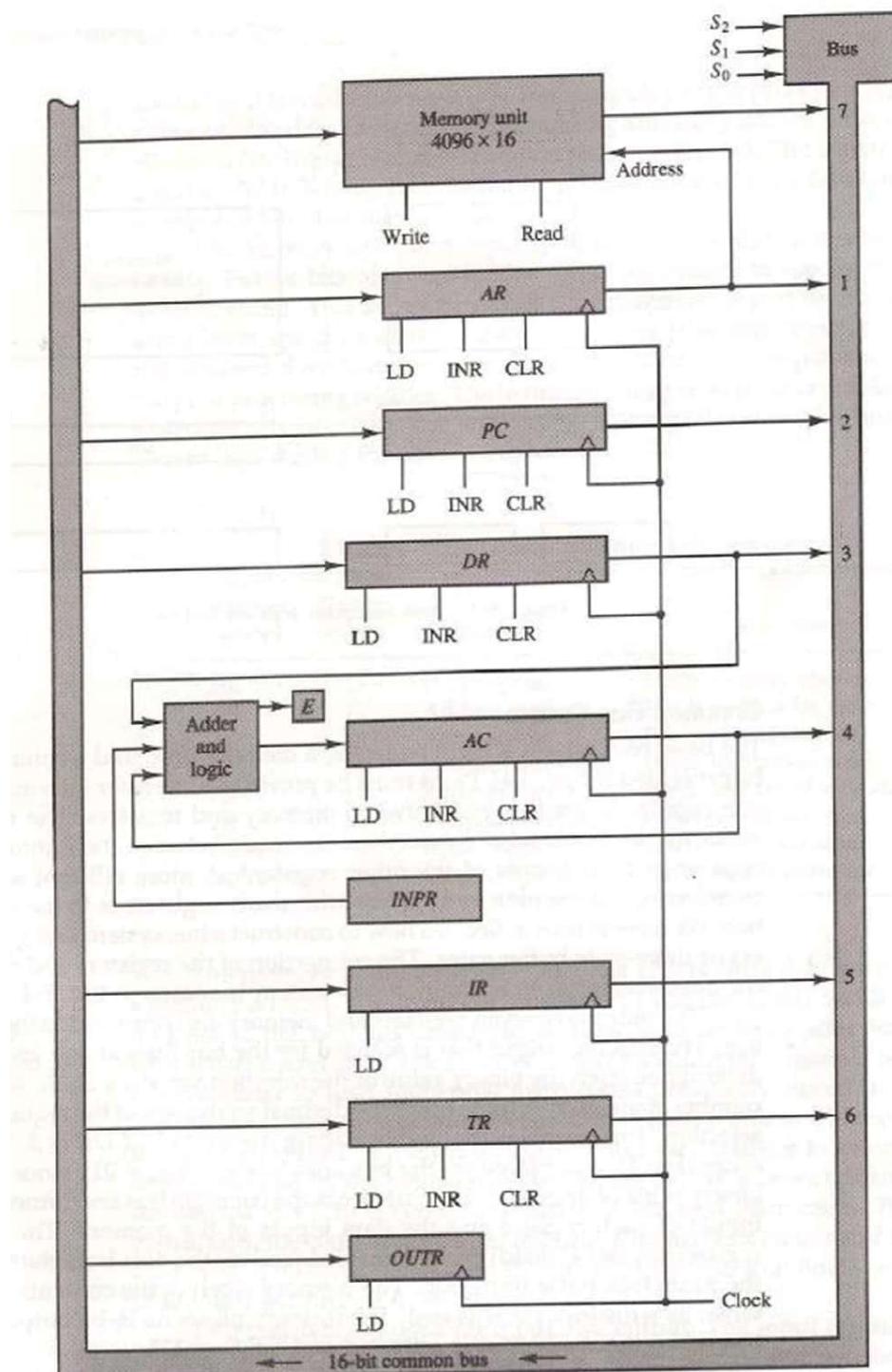


Figure 5-4 Basic computer registers connected to a common bus.

Lines from the common bus are connected to the inputs of each register and the data memory.

DATE :

SUBJECT :

Basic Computer (Design) 4Kx16

LD → load

INR → increment

CIR → clear

selection line:

Source || Select SW

INPA:

COMMON bus

control lines

Source || INPA

LD Instruction

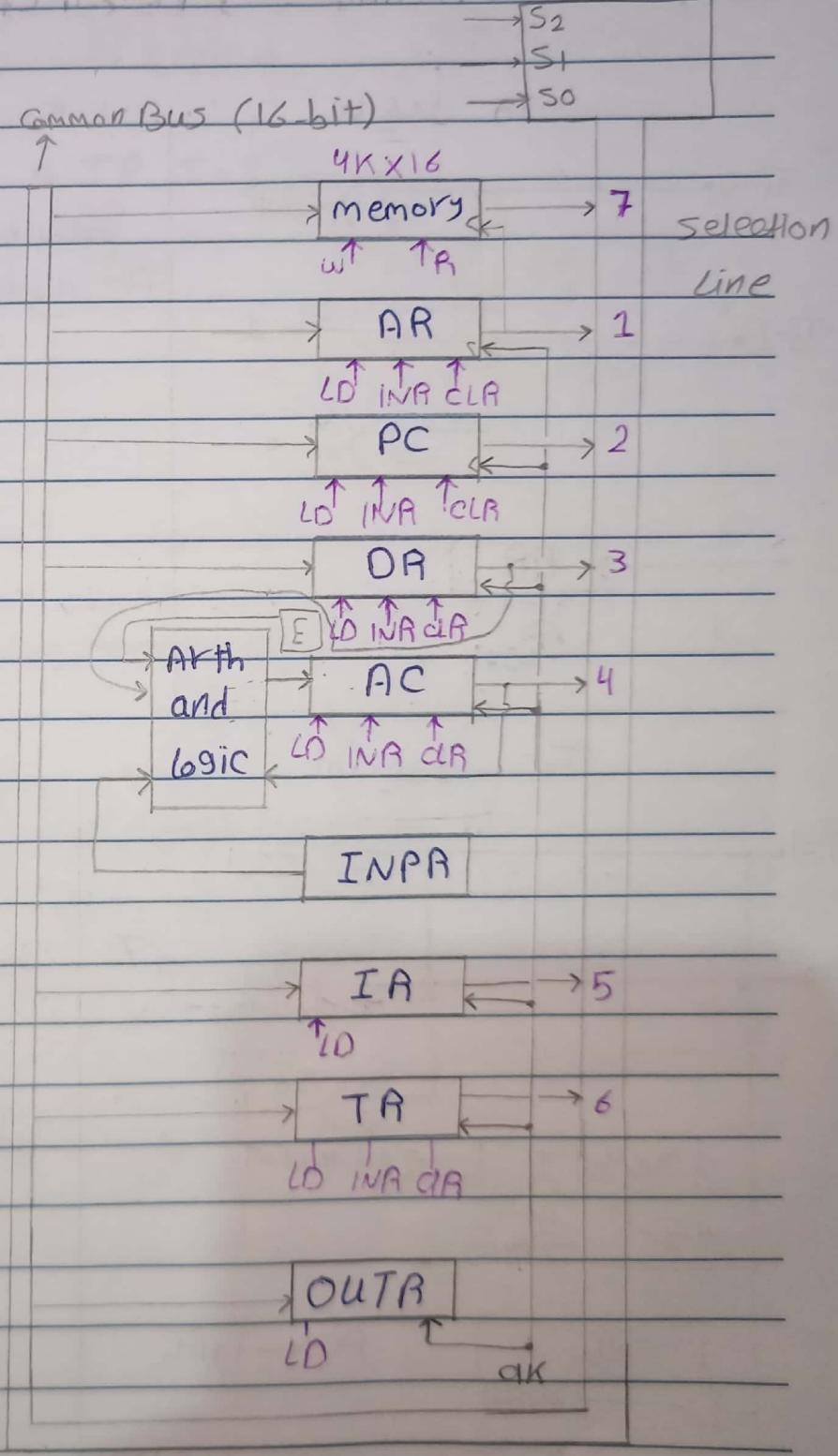
Destination || AC

* memory → source

Read || INR

* memory → destination

Write || CLR



DATE : _____

SUBJECT : _____

To: AR \leftarrow PC

Fetch

T₁: IA \leftarrow M[AR] , PC \leftarrow PC+1

To: لـ (ما يفعل)

S₂ S₁ S₀

0 1 0 \rightarrow 10

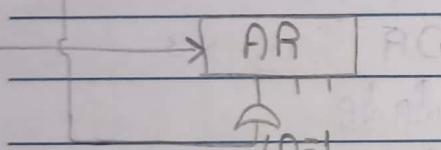
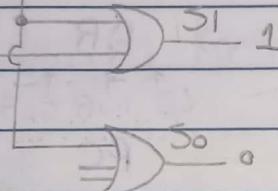
مقدمة أن إلى له الارتداد دوافع

PC هو Source of Data

T₁=0



T₀=1



T₁: ما يفعل

دالة التي لها الارتداد لتوصيف

Source S Data

وهو الذاكرة لبيانات

R=1 يضع الـ Memory

في الـ Memory لازم الاتساع ولهذا

Memory \leftarrow AR

W R1

PC \leftarrow 1+PC

INR=1

IR \leftarrow M[AR]

ALAQSA

ence counter SC is cleared to 0, providing a decoded timing signal T_0 .
 ooperations for the fetch and decode phases can be specified by the following register statements.

$$T_0: AR \leftarrow PC$$

$$T_1: IR \leftarrow M[AR], \quad PC \leftarrow PC + 1$$

$$T_2: D_0, \dots, D_7 \leftarrow \text{Decode } IR(12-14), \quad AR \leftarrow IR(0-11), \quad I \leftarrow IR(15)$$

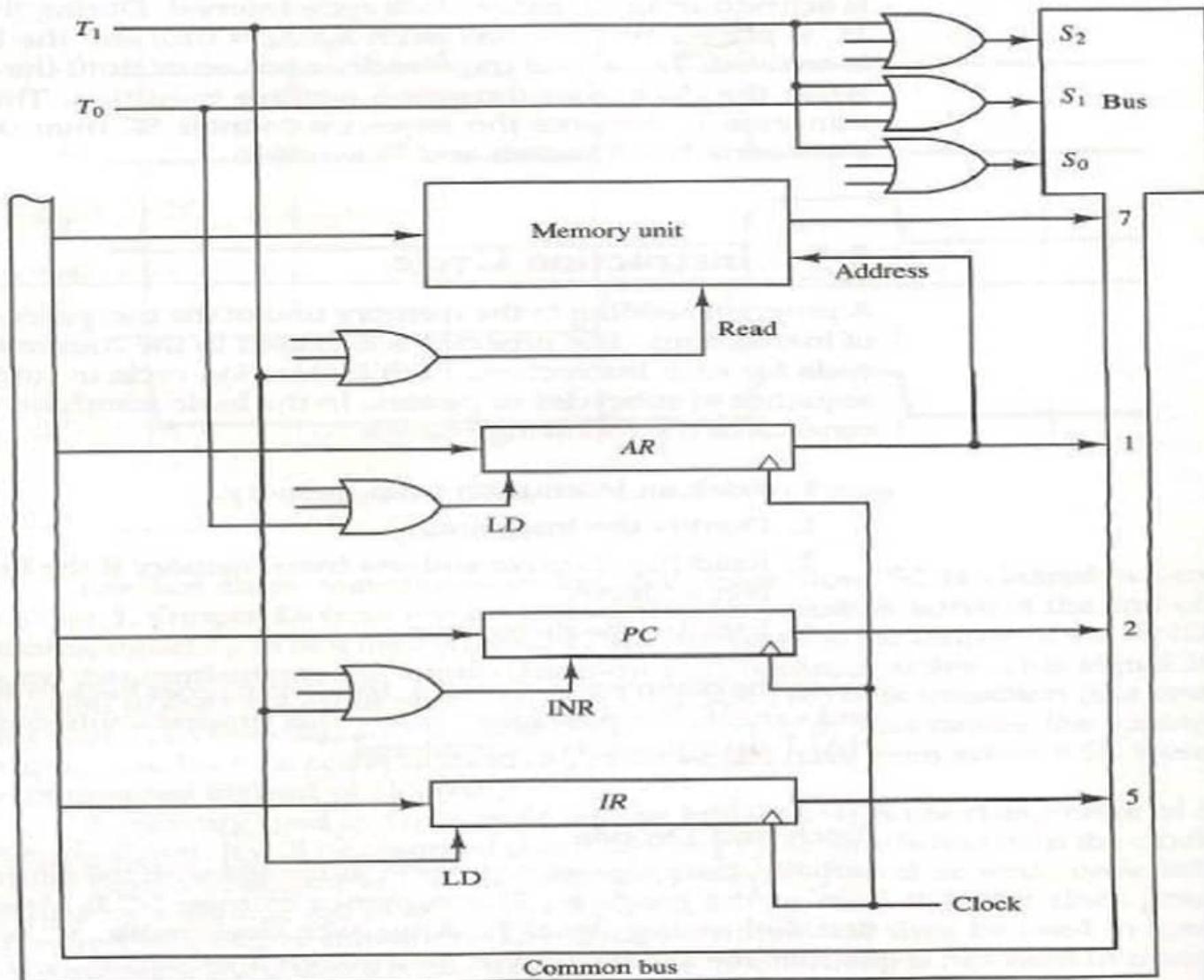


Figure 5-8 Register transfers for the fetch phase.

8 shows how the first two register transfer statements are implemented in the bus system.
 To implement the data path for the transfer of *PC* to *AR* we must apply timing signal *T₀* to achieve the required connection:

DATE :

SUBJECT :

Set of instruction

I=0 \rightarrow direct Add

I=1 \rightarrow indirect Add

OP-Code

IR | I | : | Address / Operand

IP I=0

IP I=1

AR

2F3

F2F3

AR=3FA

operand \leftarrow 3FA = AR

DR

DR

*OP Code

memory instruction reference

AND

ADD

i [] instruction

a. up to 5 operands

HLT

b. up to 2 operands

16=1011

0=1000

4K X 16

size (2^{12}) memory address, word length
 12-bit 16-bit

1- Internal Register

DR: Data reg	Word length 16-bits
TR: Temporary reg	
IR: Instruction reg	
AC: Accumulator (Processor's reg) → E (extended flag)	

AR: Address reg (memory pointer)	hold address 12 bits
PC: Program Counter	

INPA: ASCII code FGI Flag input	I-o device 8-bits
OUTA: output reg FGO Flag output	

16-bit reg يعني لو one F.F يحتاج bit 15 notes
 16-F.F 12 بت

الآن الامر انتertain ويلعب على اعلام واحد اسرا FGI=1 ←
 لاعم انتertain لـ ASCII code لـ user ما ينفعه

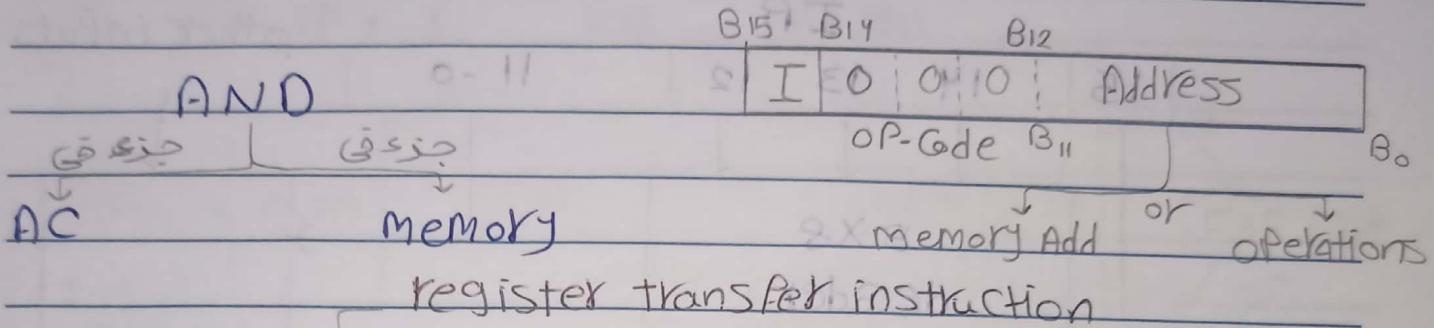
Basic computer goes low says ان الامر انتertain ويلعب على اعلام واحد FGI=0 ←
 Key board goes char يعني 001

DATE :

SUBJECT :

2. Timing and Control Structure

~~memory reference Instruction.~~



To: AR \leftarrow PC micro operation

Tutoring available at the NLRB Library

$T_1 \cdot T[G] \leftarrow M[\text{EAR}] \rightarrow PC \leftarrow PC + 1$

~~current instr~~ next instr

Symbol	Instruction	Description
	Hexa-decimal Code	
I=0		I=1
AND	0 XXXX_H	D ₀ 8 XXXX_H
ADD	1 XXXX_H	D ₁ 9 XXXX_H
LDA	2 XXXX_H	A XXXX_H
STA	3 XXXX_H	B XXXX_H
BUN	4 XXXX_H	C XXXX_H
BSA	5 XXXX_H	D XXXX_H
ISZ	6 XXXX_H	D ₆ E XXXX_H
Decoder		
3x8		

Memory reference Instruction

7

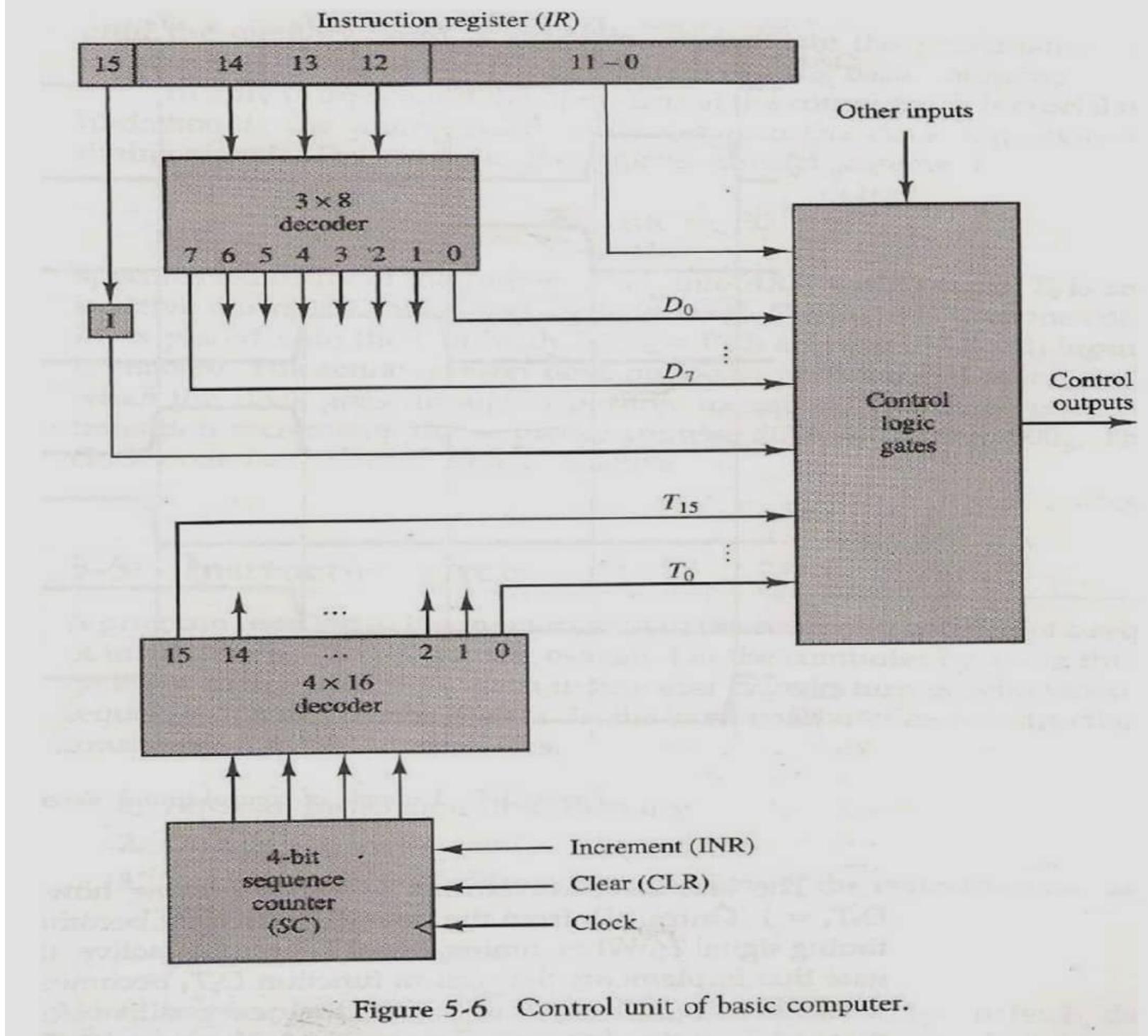
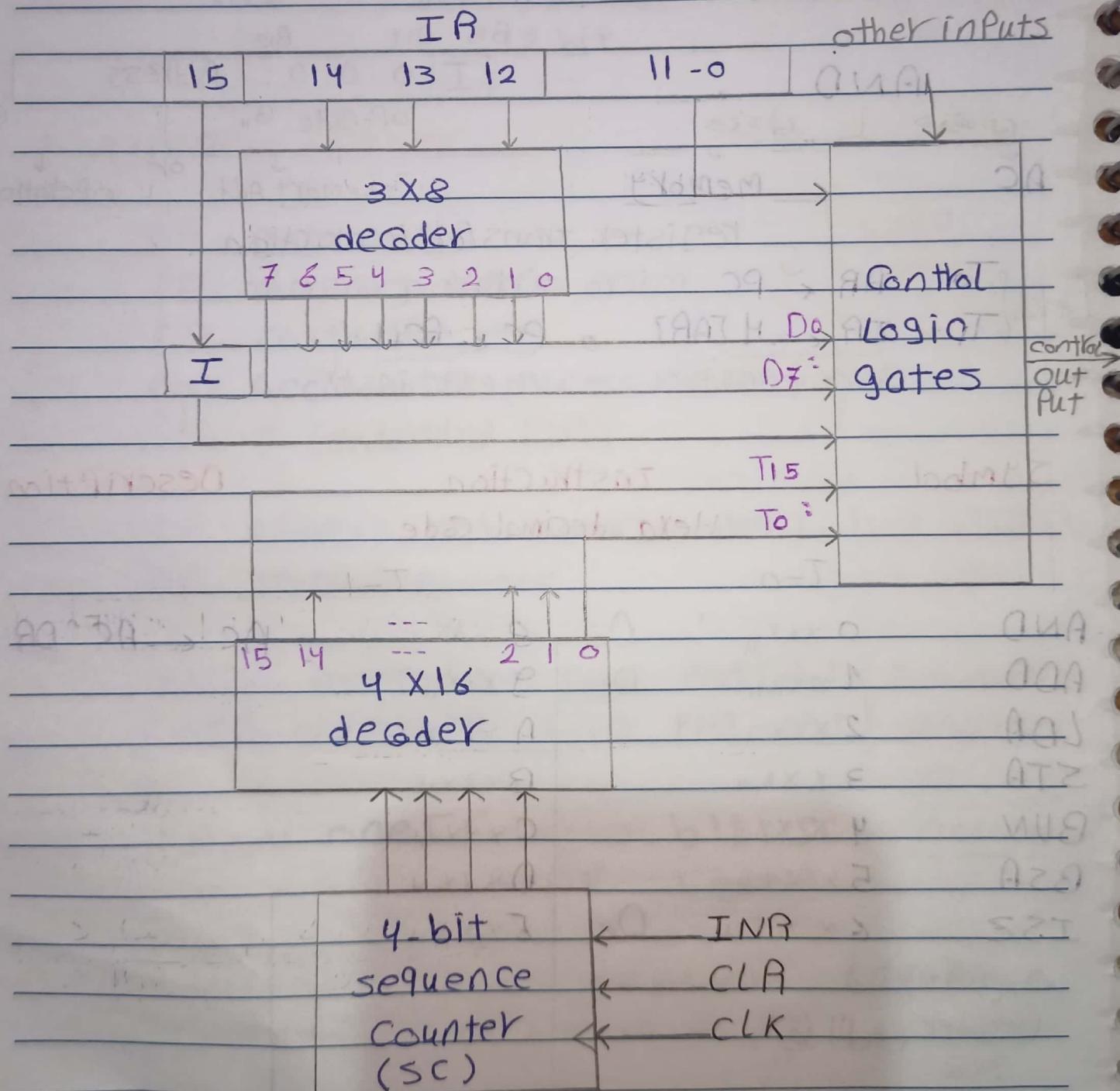


Figure 5-6 Control unit of basic computer.

SUBJECT : -

DATE : -

T₂: I \leftarrow IA[15], Decoder 3×8 IA(12-14)
AR \leftarrow IR[0-11]



* Control Unit of basic computer *

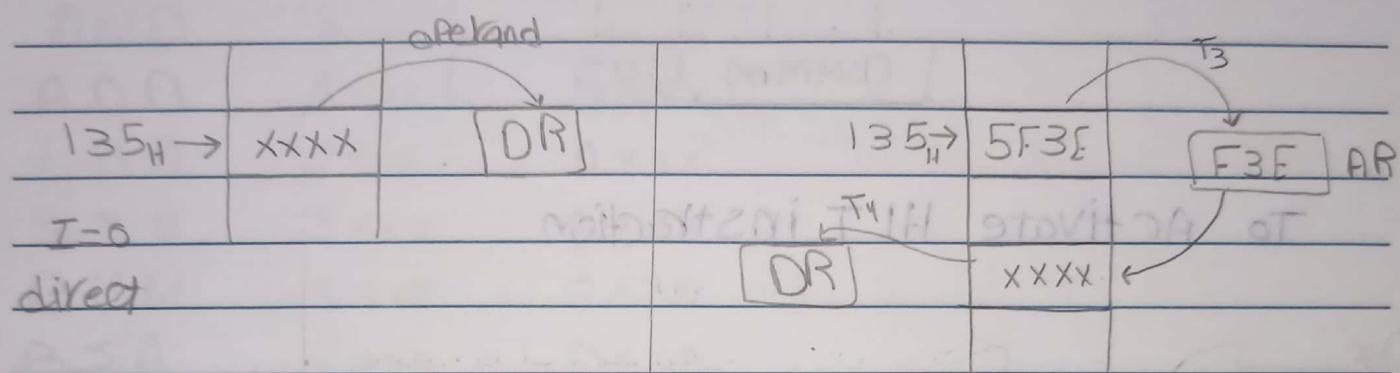
DATE : _____

SUBJECT : _____

... ADD, AND ... op-code الارقام ← (12-14 bit)

T3: IP I=0 (Nothing to Do)

IP I=1 AR ← M[AR]



T4: DR ← M[AR]

T5: AC ← DR^AC, SC=0 clear

* Register Reference Instruction.

I is always zero

OP.Gde also, 111

IR (0-11) → is operation

15 14 12
0 1 1 1 | Operation

7 قيم "I" .

CLA

7800

12 11 10 9 8 7 6 5 4 3 2 1 0

CLE

7400

0x-10000000

CMA

7200

1000000000000000

CME

7100

Add 1111111111111111 I *
direct or indirect

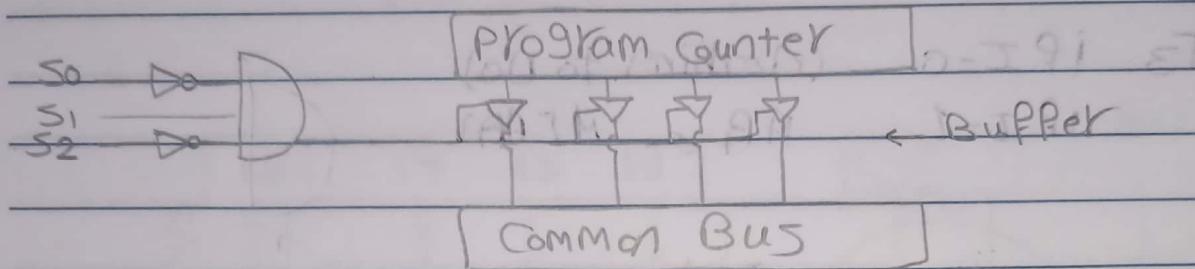
HLT

12 INSTRUCTIONS

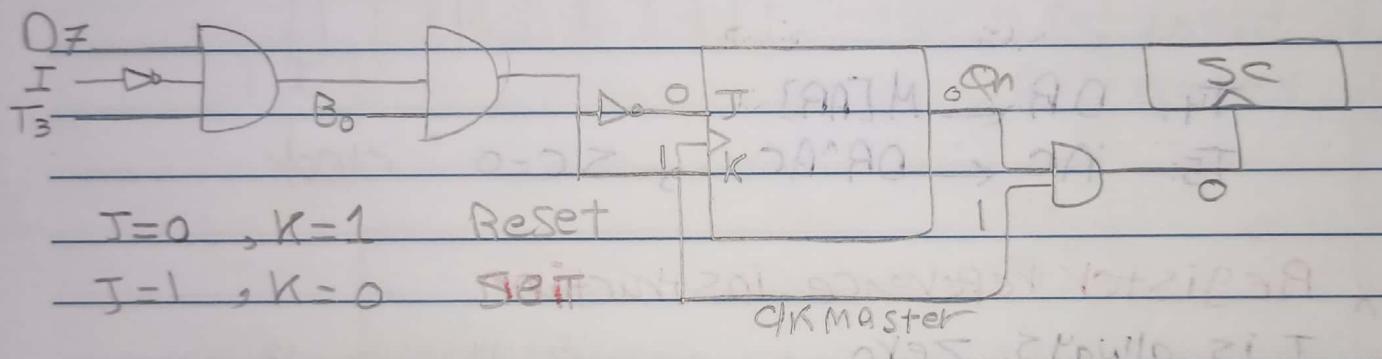
DATE : _____

SUBJECT : _____

Selection قریبی CB ای PC مواد ای ایجاد مراحل



To ACTIVATE HLT instruction.



I/O reference instructions

- I is always = 1
- OP Code 111

1 | 1 1 1 |

INP	F800
OUT	F400
\$KI	F200
\$KO	F100
ION	F080
IOFF	F040

6. Instruction

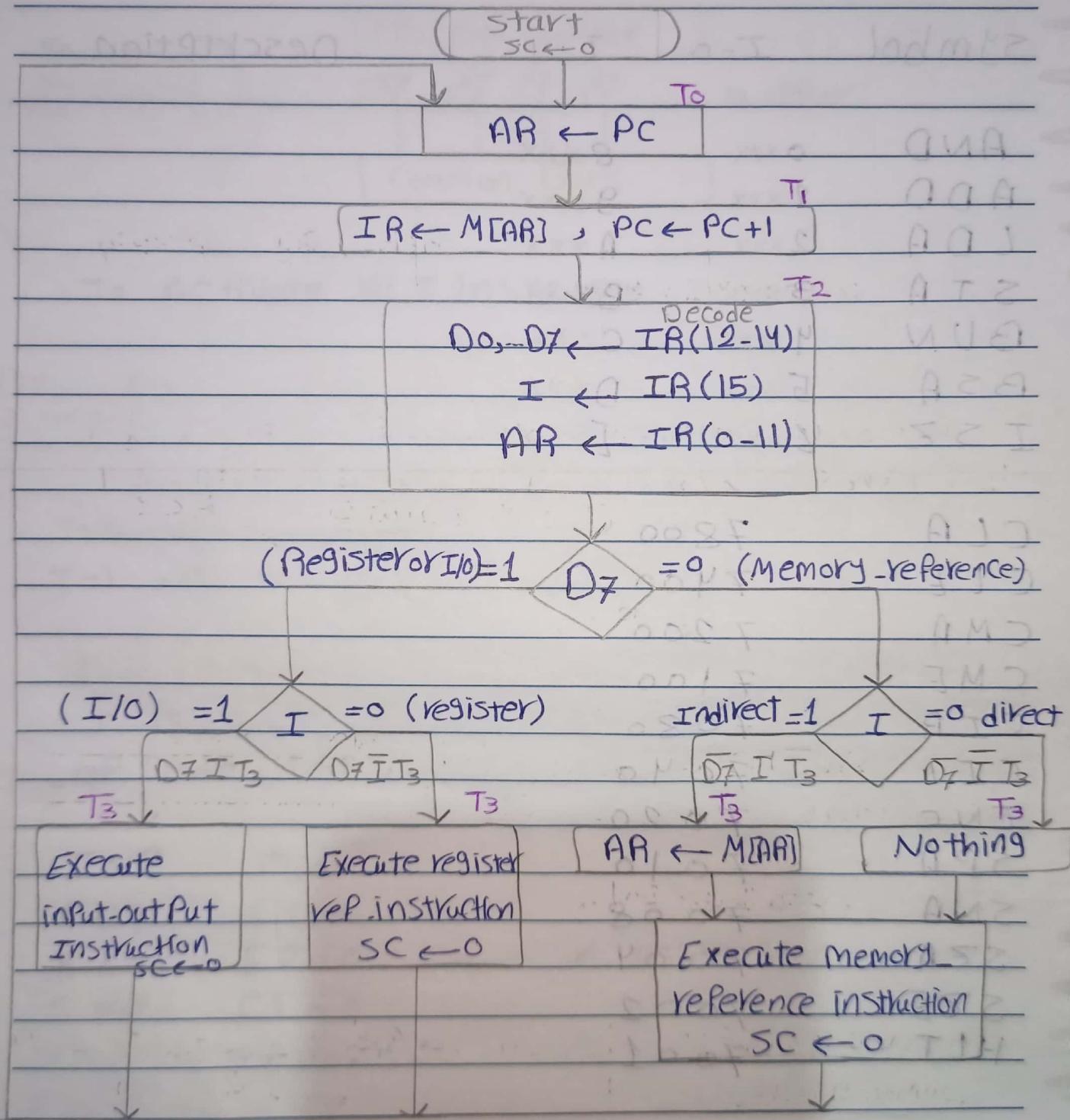
TABLE 5-2 Basic Computer Instructions

Symbol	Hexadecimal code		Description
	<i>I</i> = 0	<i>I</i> = 1	
AND	0xxx	8xxx	AND memory word to <i>AC</i>
ADD	1xxx	9xxx	Add memory word to <i>AC</i>
LDA	2xxx	Axxx	Load memory word to <i>AC</i>
STA	3xxx	Bxxx	Store content of <i>AC</i> in memory
BUN	4xxx	Cxxx	Branch unconditionally
BSA	5xxx	Dxxx	Branch and save return address
ISZ	6xxx	Exxx	Increment and skip if zero
CLA	7800		Clear <i>AC</i>
CLE	7400		Clear <i>E</i>
CMA	7200		Complement <i>AC</i>
CME	7100		Complement <i>E</i>
CIR	7080		Circulate right <i>AC</i> and <i>E</i>
CIL	7040		Circulate left <i>AC</i> and <i>E</i>
INC	7020		Increment <i>AC</i>
SPA	7010		Skip next instruction if <i>AC</i> positive
SNA	7008		Skip next instruction if <i>AC</i> negative
SZA	7004		Skip next instruction if <i>AC</i> zero
SZE	7002		Skip next instruction if <i>E</i> is 0
HLT	7001		Halt computer
INP	F800		Input character to <i>AC</i>
OUT	F400		Output character from <i>AC</i>
SKI	F200		Skip on input flag
SKO	F100		Skip on output flag
ION	F080		Interrupt on
IOF	F040		Interrupt off

DATE : Lec 3

SUBJECT :

Flowchart For instruction cycle



Control then inspects the value of the first bit of the instruction, which is now available in D_7 . If $D_7 = 0$ and $I = 1$, indicates a memory-reference instruction with an indirect address. So it is necessary to read the effective address from memory.

If $D_7 = 0$ and $I = 0$, indicates a memory-reference instruction with a direct address.

If $D_7 = 1$ and $I = 0$, indicates a register-reference instruction.

If $D_7 = 01$ and $I = 1$, indicates an input-output instruction.

The three instruction types are subdivided into four separate paths.

The selected operation is activated with the clock transition associated with timing signal T . This can be symbolized as follows:

- $D_7' IT_3: AR \leftarrow M[AR]$
- $D_7' I'T_3: \text{Nothing}$
- $D_7 I'T_3: \text{Execute a register-reference instruction}$
- $D_7 IT_3: \text{Execute an input-output instruction}$

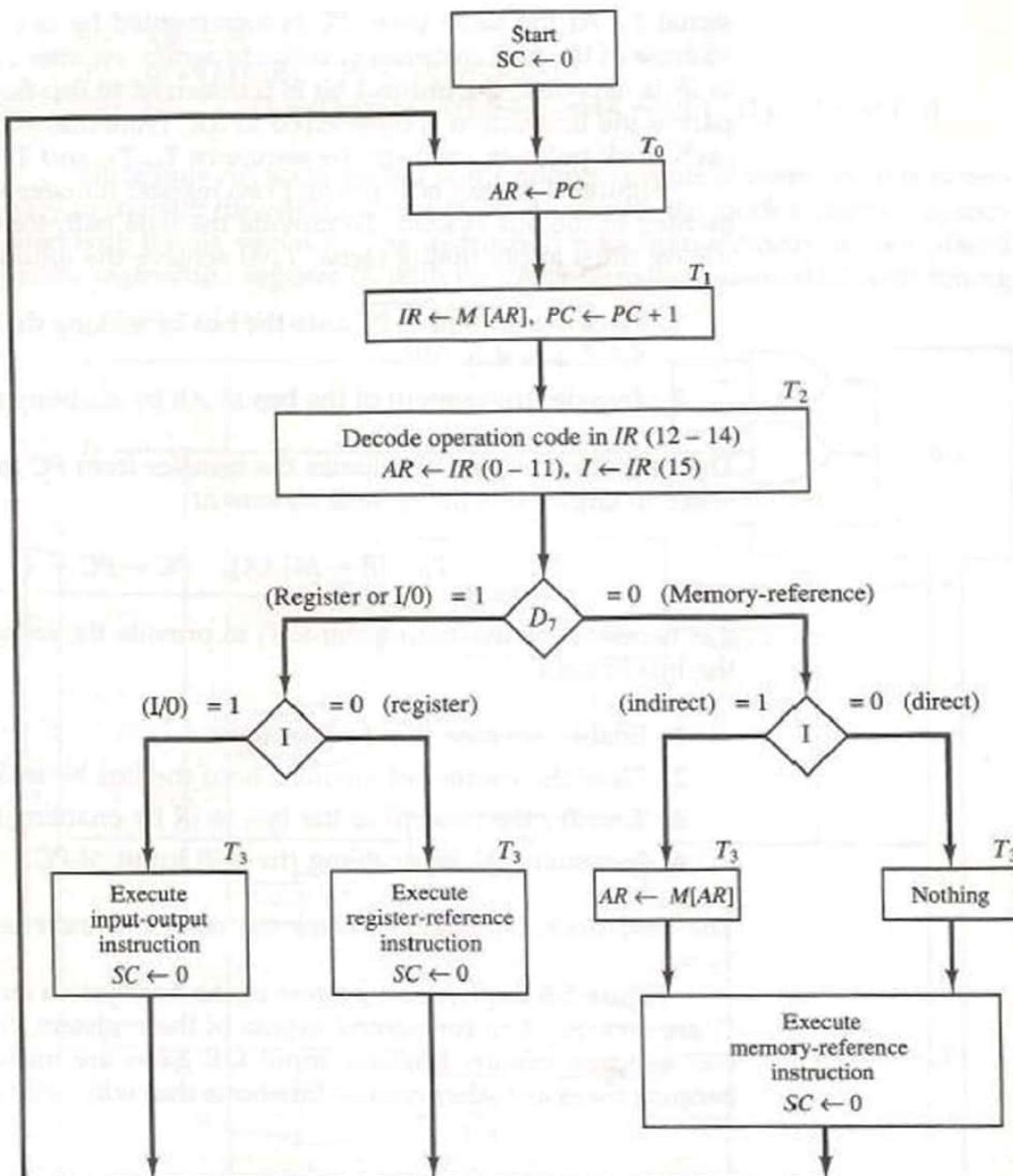


Figure 5-9 Flowchart for instruction cycle (initial configuration).

* Determine the type of Instruction:

- memory reference instr
- register " "
- O/I " "

* memory reference instruction:

1) AND: 0000 or 1000

D₁T₄: DR ← M[AR]

D₁T₅: AC ← AC ^ DR, SC ← 0

DR ← M[AR] (Memory location address) ← Address Register (AR)

AC ← AC ^ DR (AC) ← Adder Unit (AU)

2) ADD: 0001 or 1001

D₁T₄: DR ← M[AR]

D₁T₅: AC ← AC, DR, E ← cout, SC ← 0

AC ← AC (Accumulator) + DR (Register) ← Adder Unit (AU)

Ebit ← "extend of acc" ← خارج من acc (end arry) ← طرفی (arry)

3) LDA: 0010 or 1010

load to Acc

D₂T₄: DR ← M[AR]

D₂T₅: AC ← DR, SC ← 0

DR ← M[AR] (Memory location address) ← Address Register (AR)

AC ← DR (Register) ← Adder Unit (AU)

Raw signal from bus Adder section and logic circuit

DATE :

SUBJECT :

4) STA: direct indirect

Store ACC

D₃T₄: M[AR] ← AC, SC ← 0

نحوه متغير الميموري داخل Acc لوصول اى ميموري وتحتاج لوسیط
COMMON bus بار AC مدخل

5) BUN: 0100 or 1100 Branch unconditional

D₄T₄: PC ← AR, SC ← 0

instr (current instr) → current instr

BUN

* بعد ما أذاع برمجية وفتح المدخلات والخرجات على المدخلات والخرجات

6) BSA: Branch and Save return address

T₄D₅: M[AR] ← PC, AR ← AR + 1

T₅D₅: PC ← AR, SC ← 0

Direct Add

↑ Memory

020H	BSA	135H	Current Instr
021H			← next Instr

next PC على الميموري
فهذا ينبع من الميموري

135H معاه فـ 1 BUN 135

وتناول على الميموري الى جوا 021H
ونلاحظ في الـ PC ويعود من الميموري

بس لو يصفر او يرجع الى 135H

ويكرر الـ procedure يعني

loop عمل

135H	021H	return add M[AR] ← PC
136H	Procedure	AR ← AR + 1
(1) BUN 135		PC ← AR

DATE :

SUBJECT :

IS Z. (Increment and skip if zero)

D₆T₄: DR \leftarrow M[AR]

D₆T₅: DR \leftarrow DR + 1

D₆T₆: M[AR] \leftarrow DA, IF (DR = 0) then PC \leftarrow PC + 1, SC \leftarrow

FFFF DR \leftarrow M[AR]

1+ DR \leftarrow DR + 1

0000 DR = 0

1+ PC \leftarrow PC + 1

0001

يعني إذا زéro و واحد والباقي الاسترجاع حفظ

Skip if زéro المفترض واحد

* Design Write Memory.

destinations يذهب إلى write memory ||| مرام

Source الذي يذهب إلى Read طورين

* المهم بنتوك في الجدول ما هي التي في الاعتبار كل الاوامر

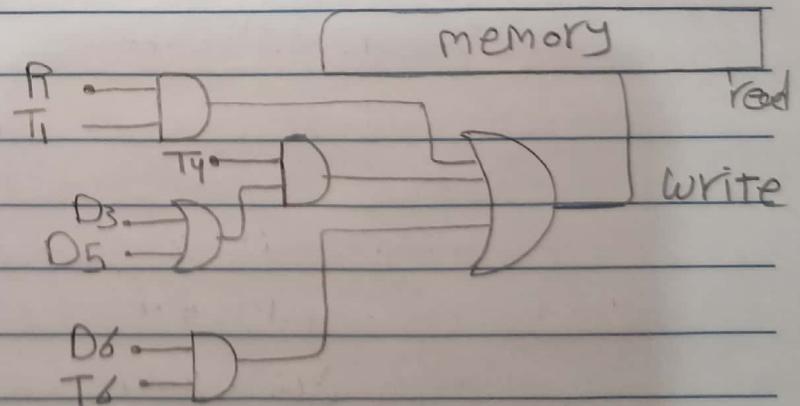
وبنطريق design already done destination في write memory |||

. R T₁ : MEAR \leftarrow TA

. D₃T₄ : MEAR \leftarrow AC

. D₅T₄ : MEAR \leftarrow PC

. D₆T₆ : MEAR \leftarrow DR



write: R T₁ + (D₃ + D₅) T₄ + D₆T₆

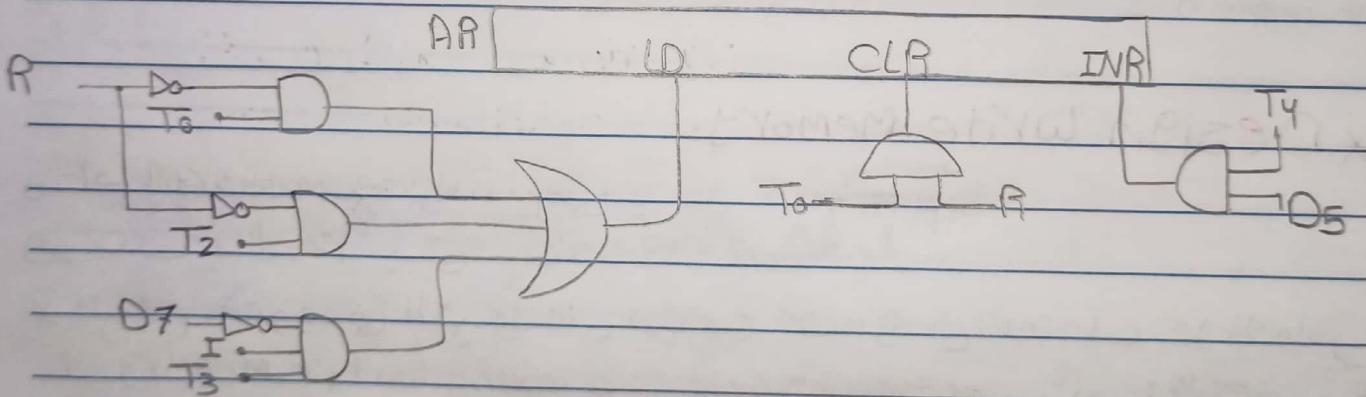
DATE : _____

SUBJECT : _____

Design address register [AR]

نحوه الوجه (destination) AB نحوه الماء (object) CD

- R T₀: AR ← PC
 - R T₂: AR ← IR(0 II)
 - D₇ IT₃: AR ← M[AR]
 - R T₀: AR ← 0 → clear
 - D₅ T₄: AR ← AR₁ → INR



Interrupt flag:

يحصل في اي قوه زئيفه على T_0, T_1, T_2 عما يحصل في الـ Fetch, decode

\rightarrow Interrupt استجابة لـ ~~استجابة~~ مؤشر (FGI) \leftarrow $ACC \cup$ انتقال وعاليز character لوعند *

$$AC(0-7) \leftarrow INPA$$

(جاء) امر لستة ترتيب $FGI = 1$ او (تم) استقبل الـ char $FGI = 0$

* لوعند BASIC computer في الـ AC وعاليز اطاعها الـ char

$$OUTR \leftarrow AC(0-7)$$

$$\text{لهم سبقني} FGO = 1$$

$$\text{استقبل خارجا} FGO = 0$$

* ممكن يحصل لو في امر ذي الـ IEN او اي امر اخر كتبته في الكور

فيمكن مقاطعه على حسب الكور اللي انت كتبته

* $R = 1$ او $R = 0$ حاله $FGI = 1$

\rightarrow R Flag Register

$$1 = FGO \text{ او } FGI , \quad \bar{T}_0 \cdot \bar{T}_1 \cdot \bar{T}_2 = 1 \text{ و}$$

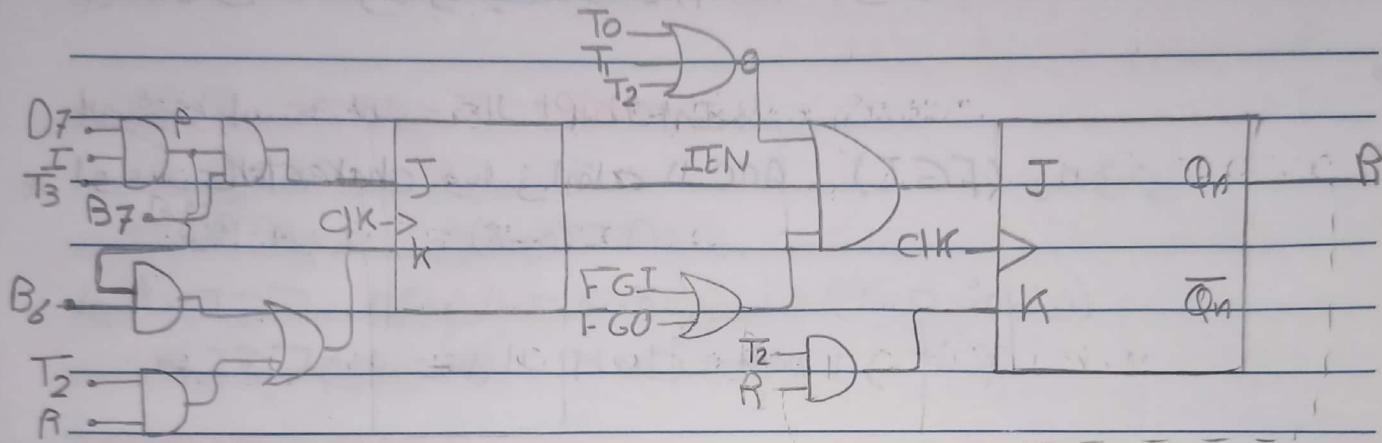
$$R = IEN \cdot \bar{T}_0 \cdot \bar{T}_1 \cdot \bar{T}_2 \cdot (FGI + FGO) \quad \times \times$$

$$(T_0 + T_1 + T_2)$$

DATE : _____

SUBJECT :

Design Interrupt cycle.



أولى الخطوات في تصميم المطبخ هي تحديد المساحة المتاحة.

$R \leftarrow 0$ لـ c_1 في الـ T_2 ، c_2 في الـ T_1 ، c_3 في الـ T_3

* امار ایه حوار ای ENI و اید موجوونی الجیول

$IEN \leftarrow 0$ or $IEN \leftarrow 1$

نحوه (ج) (الحالات الظاهرة في الجملة) يسمى بالـ **الحال**.

محلات آسفي (P) لـ تور جبوا

D₇ I T₃ - P (common to all input-output instr)

لهم ايه حلم دعيتني لازم تدخل في الريانين بسأعلم

PBF لينا TENK 1 ورنا \rightarrow T حلا

وعن الـ IEC 60068-2-27 معايير مقاييس واصف في

وواد٢٥٠ RT₂ ملائين على RT₂ وس

* طبیعت کتابت فی طبیعت زی الـ م اسـهـا ۷ بـرـاهـهـ زـرـیـهـاـ وـ لـازـمـهـ تـدـلـیـفـیـ
الـبـیـزـلـیـنـ

DATE : _____ SUBJECT : _____

Interrupt Cycle -

وجوده برمي الجدول
الجدول يفتح طول المدة كـ

RT₀: AR ← 0 , TR ← PC

Memory location (AR) يعطى AR || CIR value

next instrr لـ عنوان PC أو العنوان الذي يحتوي على PC
نفس المدة هنا في المدة المدورة

تحدد المدة التي تأخد TR لـ

RT₁: M[AR] ← TR , PC ← 0

وتحدد العنوان في المدة المدورة

RT₂: PC ← PC + 1 , IEN ← 0 , R ← 0 , SC ← 0

لـ أول إجراء

وتحدد باقى اجراءات المعاشر

$$5x_1 + 2x_2 + 8x_3 + 1x_4 = 07$$

$$5x_1 + 2x_2 + 5x_3 + 5x_4 = 12$$

$$5x_1 + 2x_2 + 3x_3 + 1x_4 = 03$$

DATE : _____

SUBJECT : _____

Design Common Bus.

encoder I/P								O/P			Selected Reg
X_0	X_1	X_2	X_3	X_4	X_5	X_6	X_7	S_2	S_1	S_0	
0	0	0	0	0	0	0	0	0	0	0	NONE
0	1	0	0	0	0	0	0	0	0	1	AR
0	0	1	0	0	0	0	0	1	0	0	PC
0	0	0	1	0	0	0	0	1	1	0	DR
0	0	0	0	1	0	0	0	1	0	0	AC
0	0	0	0	0	1	0	0	1	0	1	IR
0	0	0	0	0	0	1	0	1	1	0	TR
0	0	0	0	0	0	0	1	1	1	1	Memory

الناتج من المدخلات المفتوحة (Common bus) موجود في الرسم
 هو على أرقام من 1 إلى 7 ← رقم يدل على حاجة
 * يستخدم 7 من خطوط decoder أو encoder
 * Regs 3, 5, 51, 52 التي يصدرها هي
 او الصيودي

* نعمل بلوحة التحكم بوادي طلائع

$$S_0 = X_1 + X_3 + X_5 + X_7$$

$$S_1 = X_2 + X_3 + X_6 + X_7$$

$$S_2 = X_4 + X_5 + X_6 + X_7$$

* يعين هنا كل X سو فها بتشار على انه
 Source ودور عليه في الجدول سبب وفق الحالات

DATE :

SUBJECT :

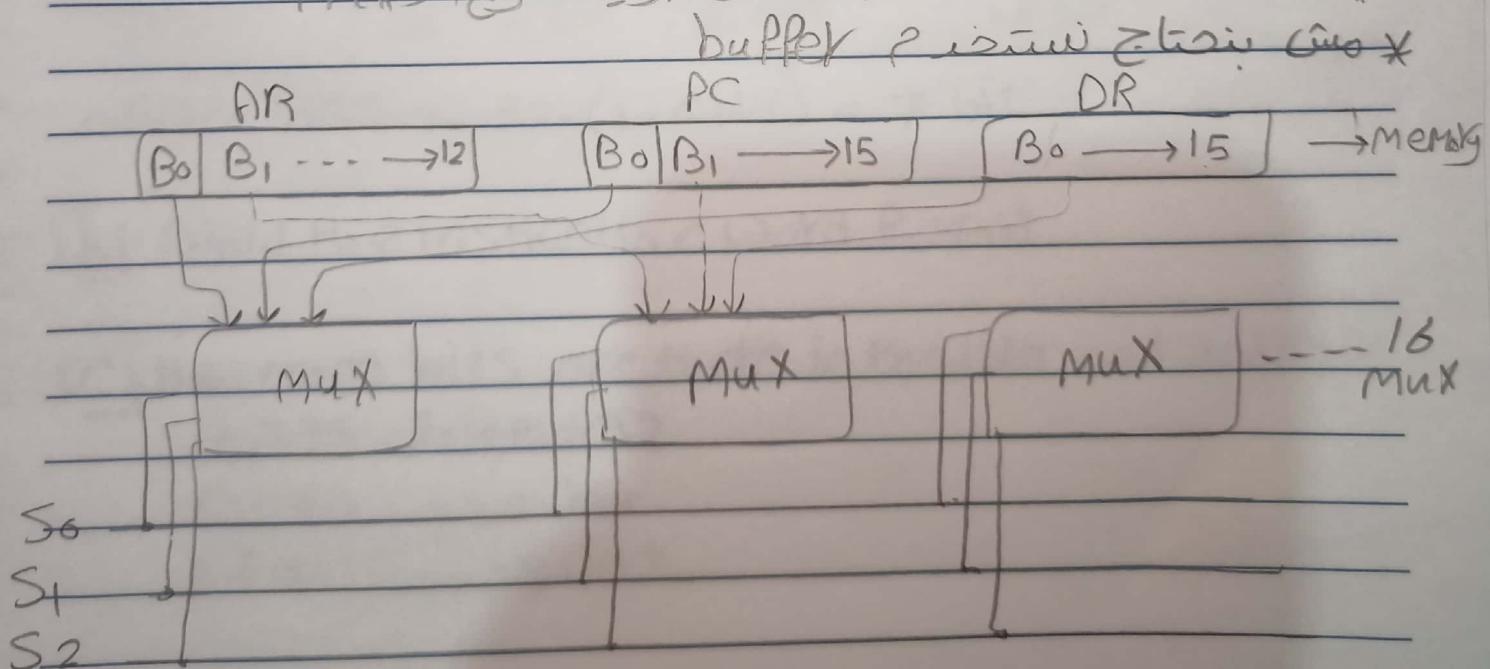
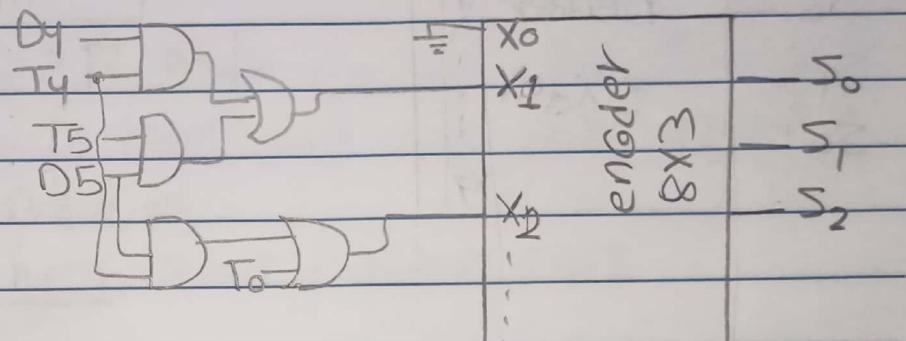
$$Y_1 = D_4 T_4 + D_5 T_5 \quad (\text{AB})$$

$$X_2 = \bar{R} T_0 + R T_0 + D_5 T_4 \quad (\text{PC})$$

$$= T_0 + D_{STY}$$

~~Source~~ 5
Guy

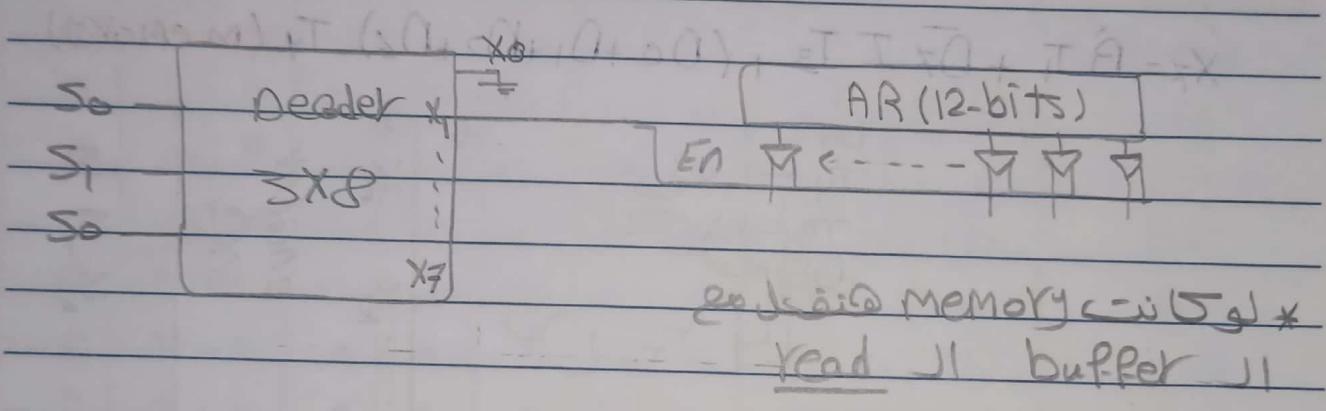
$$X_7 = \bar{D}_1 T_1 + \bar{D}_7 IT_3 + (D_0 + D_1 + D_2 + D_6) \cdot T_4 \text{ (Memory)}$$



DATE :

SUBJECT :

Mux or J or L deoder - busines
اللى عرض bits اجدر او buffers وفقاً لـ AR 12-bit 12-buffer



DATE :

Sheet

SUBJECT :

[I] A Computer uses a memory unit with 256K words of 32-bit each. A binary Instruction Code is stored in one word of memory. The instruction has 4 parts: an Indirect bit, an Opertion Code, a Register Code Part to specify one 64 registers and an address Part

[a] How many bits are there in the operation code, the register code part and the address part?

$$256 \text{ K} \rightarrow 2^{18} \rightarrow 18\text{-bit address Bus}$$

$$\downarrow 8 \quad \downarrow 10 = 2^6 \rightarrow 6\text{-bit}$$

* word length = 32-bit COMMON BUS

64 registers $\rightarrow 2^6 \rightarrow 6\text{-bit}$

[b]

I	OP-Gde	RegisterCode	address	32-bit
1-bit	7-bit	6-bit	18-bit	

$$\text{OPGde bits} = 32 - (1 + 6 + 18) = 7\text{-bit}$$

[b] Draw the instruction word Format

[c] How many bits are there in the data and address inputs of memory?

Data \rightarrow 32-bit

Address \rightarrow 18-bit

DATE :

SUBJECT :

[3] Specify the register transfer that will be executed during the next clock transition

Source J1 J2 J3 destination J1 J2 J3

S₂ S₁ S₀ LD of Register Memory Adder

a-	1 1 1	IR	Read	-
b-	1 1 0	PC	-	-
c-	1 0 0	DR	Write	-
d-	0 0 0	AC	-	Add

[a] IR ← M[AR]
read 1101010000000000
memory 7 111

[b] PC ← TR
TA ←⁶ 110

[c] M[AR] ← AC
DR ← M[AR]
AC ←⁴ 100
AC ←⁴ 100

[d] DR ← M[AR]
AC ← AC + DR , E ← C(i+1)

DATE : _____

SUBJECT : _____

4]

اوامر دالة الارجل لـ ALU وload memory

Reg transfer statement	S_2	S_1	S_0	load	Memory	Adder
	source			destination		
a- $AR \leftarrow PC$	0	1	0	AR		
b- $IR \leftarrow M[AR]$	1	1	1	IR	read	
c- $M[EAR] \leftarrow TR$	1	1	0		write	
d- $AC \leftarrow DR$, $DR \leftarrow AC$	1	0	0	AC and DR		DR transfer to AC

5]a) $IR \leftarrow M[PC]$ XX

SS اب اخراج

RT₀: $AR \leftarrow PC$ RT₁: $IR \leftarrow M[AR]$ b) $AC \leftarrow AC + TR$ XXD₁T₄: $DR \leftarrow TR$ D₁T₅: $AC \leftarrow AC + DR \rightarrow E \leftarrow (G_i + 1) \rightarrow SC \leftarrow 0$