Computer Organization

Recitation 1

Question 1

You are given the circuit in Figure 1 and asked to determine its function in terms of inputs **A**, **B**, **S0**, and **S1**. The circuit consists of a 4x1 MULTIPLEXER and a FULL ADDER. The gate implementations and representations of the 4x1 MULTIPLEXER and the FULL ADDER are given in Figure 3 and Figure 4, respectively.

Please fill in the table in Figure 2 with the function of the circuit which is obtained by means of **S** output of the FULL ADDER (e.g., **assuming you think that** the circuit output **S = A'+ B'** when **S0=0** and **S1=0**, you have to fill in the first row of the table with the text **A' + B'**).

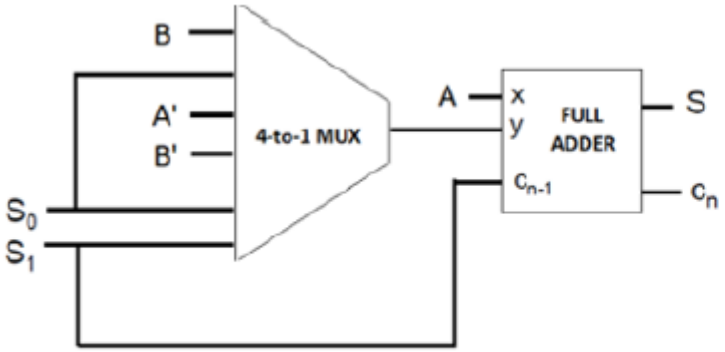


Figure 1: The Given Circuit

S_1 S_0	S (OPERATION IN TERMS OF A AND B INPUT)
0 0	
0 1	
1 0	
1 1	

Figure 2: Characteristic table to be filled in

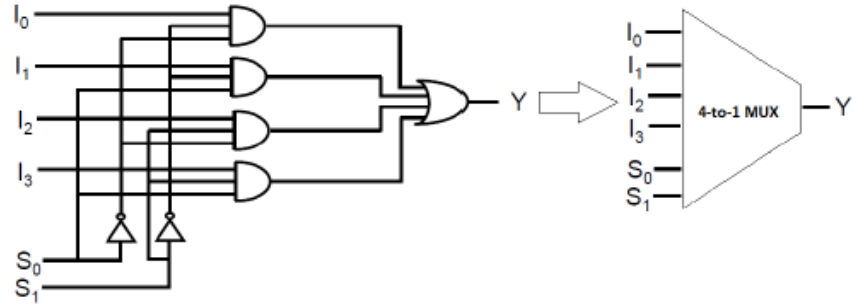


Figure 3: 4x1 Multiplexer Internals and Representation

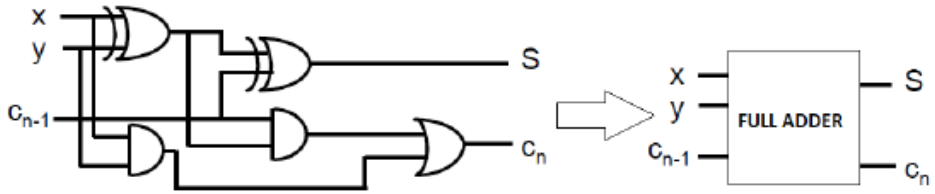
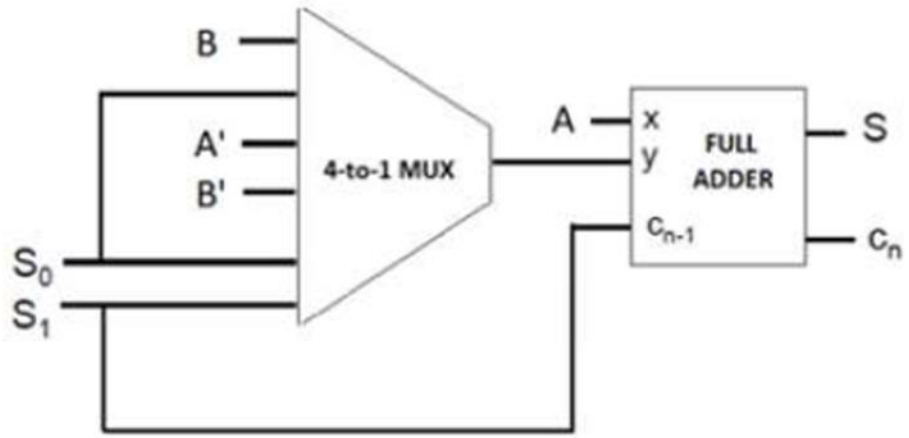


Figure 4: Full Adder Internals and Representation

Solution 1

Inputs of FULL ADDER are: x = A, y = Selected input by MUX(S_0,S_1), and $C_{n-1} = S_1$.



S_1 S_0	S (OPERATION IN TERMS OF A AND B INPUT)
0 0	S = A + B + 0 => S = A + B
0 1	S = A + 0 + 1 => S = A + 1
1 0	S = A + A' + 1 => S = A – A => S = 0
1 1	S = A + B' + 1 => S = A - B

For S1=0 and S0=0, inputs of FULL ADDER are: A, B, 0 => S = A + B + 0 => **S = A + B**

For S1=0 and S0=1, inputs of FULL ADDER are: A, 1, 0 => S = A + 0 + 1 => **S = A + 1**

For S1=1 and S0=0, inputs of FULL ADDER are: A, A', 1 => S = A + A' + 1 => **S = A – A => S=0**

For S1=1 and S0=1, inputs of FULL ADDER are: A, B', 1 => S = A + B' + 1 => **S = A - B**

Question 2

The content of the memory for the basic computer, as explained in lectures, is given along with the content of PC, AC and E in Fig.1. Assuming that the basic computer starts with fetching an instruction,

a) Decode and show the first instruction to be fetched.

b) Execute the instruction and show the contents of

i. PC and IR,

ii. AR and I,

iii. AC, E, and DR,

c) Proceed with fetching the next instruction and apply steps a) and b).

(Hint: It is suggested to create a table similar to the one in Fig.2 and fill for each operational step. Also relevant computations have to be shown clearly.)

PC	AC	E	IR	I	AR	DR	Operation / Explanation
01AB	34FF	1					

Fig 2. A typical table to fill for each operational step

Memory

000	7777
:	:
1AA	83A1
1AB	91FF
1AC	3201
1AD	12F0
1AE	C29F
1AF	929F
1B0	7001
:	:
1FF	92F0
200	32C8
201	02F0
:	:
2F0	829F
:	:
29F	7001
:	:
3A1	0200
:	:
FFF	0000

PC	01AB
AC	34FF
E	1

Fig 1. Initial content of some memory cells, registers and a flag

Solution 2

CLOCK CYCLE	PC	AC	E	IR	I	AR	DR	OP	EXP
T= 0	1AB	34FF	1			1AB			AR ← PC
T=1	1AC			91FF					IR ← M[AR], PC ← PC +1
T=2					1	1FF		ADD	AR←IR(0-11), I ←IR(15), Decode IR(12-14)
T=3					1	2F0			AR ← M[AR]
T=4							829F		DR ← M[AR]
T=5		B79E	0						AC ← AC + DR (AC←34FF+829F), E ← Cout

CLOCK CYCLE	PC	AC	E	IR	I	AR	DR	OP	EXP
T= 0						1AC			AR ← PC
T=1	1AD			3201					IR ← M[AR], PC ← PC +1
T=2					0	201		STA	AR←IR(0-11), I ←IR(15), Decode IR(12-14)
T= 3									Nothing
T=4									M[AR] ← AC (M[201] ← B79E)

Symbol	Hex Code I = 0	Hex Code I = 1	Description
AND	0xxx	8xxx	AND memory word to AC
ADD	1xxx	9xxx	Add memory word to AC
LDA	2xxx	Axxx	Load AC from memory
STA	3xxx	Bxxx	Store content of AC into memory
BUN	4xxx	Cxxx	Branch unconditionally
BSA	5xxx	Dxxx	Branch and save return address
ISZ	6xxx	Exxx	Increment and skip if zero

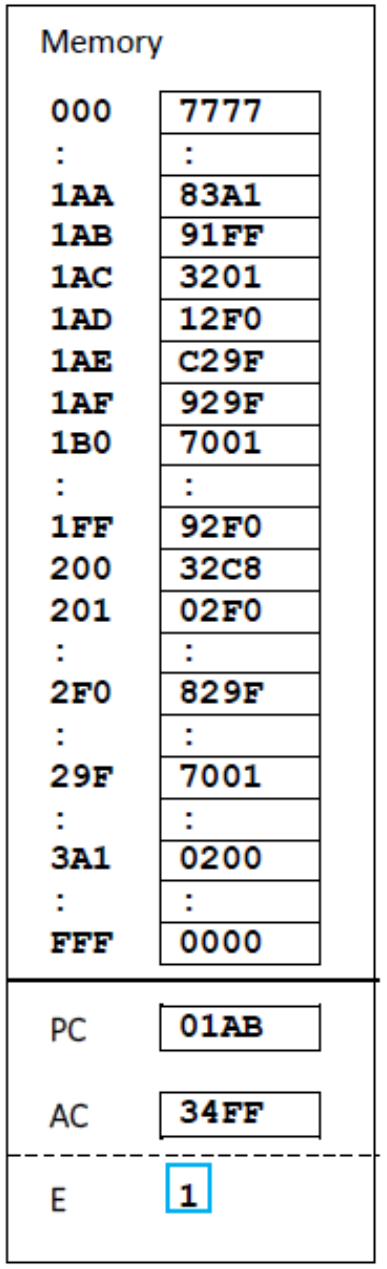


Fig 1. Initial content of some memory cells, registers and a flag

Solution 2

CLOCK CYCLE	PC	AC	E	IR	I	AR	DR	OP	EXP
T= 0						1AD			AR ← PC
T=1	1AE			12F0					IR ← M[AR], PC ← PC +1
T=2					0	2F0		ADD	AR←IR(0-11), I ←IR(15), Decode IR(12-14)
T=3									Nothing
T=4							829F		DR ← M[AR]
T=5		3A3D	1						AC ← AC + DR (AC←B79E+829F), E ← Cout

CLOCK CYCLE	PC	AC	E	IR	I	AR	DR	OP	EXP
T= 0						1AE			AR ← PC
T=1	1AF			C29F					IR ← M[AR], PC ← PC +1
T=2					1	29F		BUN	AR←IR(0-11), I ←IR(15), Decode IR(12-14)
T=3						001			AR ← M[AR]
T=4	001								PC←AR

Symbol	Hex Code I = 0	Hex Code I = 1	Description
AND	0xxx	8xxx	AND memory word to AC
ADD	1xxx	9xxx	Add memory word to AC
LDA	2xxx	Axxx	Load AC from memory
STA	3xxx	Bxxx	Store content of AC into memory
BUN	4xxx	Cxxx	Branch unconditionally
BSA	5xxx	Dxxx	Branch and save return address
ISZ	6xxx	Exxx	Increment and skip if zero

Memory

000	7777
:	:
1AA	83A1
1AB	91FF
1AC	3201
1AD	12F0
1AE	C29F
1AF	929F
1B0	7001
:	:
1FF	92F0
200	32C8
201	02F0
:	:
2F0	829F
:	:
29F	7001
:	:
3A1	0200
:	:
FFF	0000

PC	01AB
AC	34FF
E	1

Fig 1. Initial content of some memory cells, registers and a flag

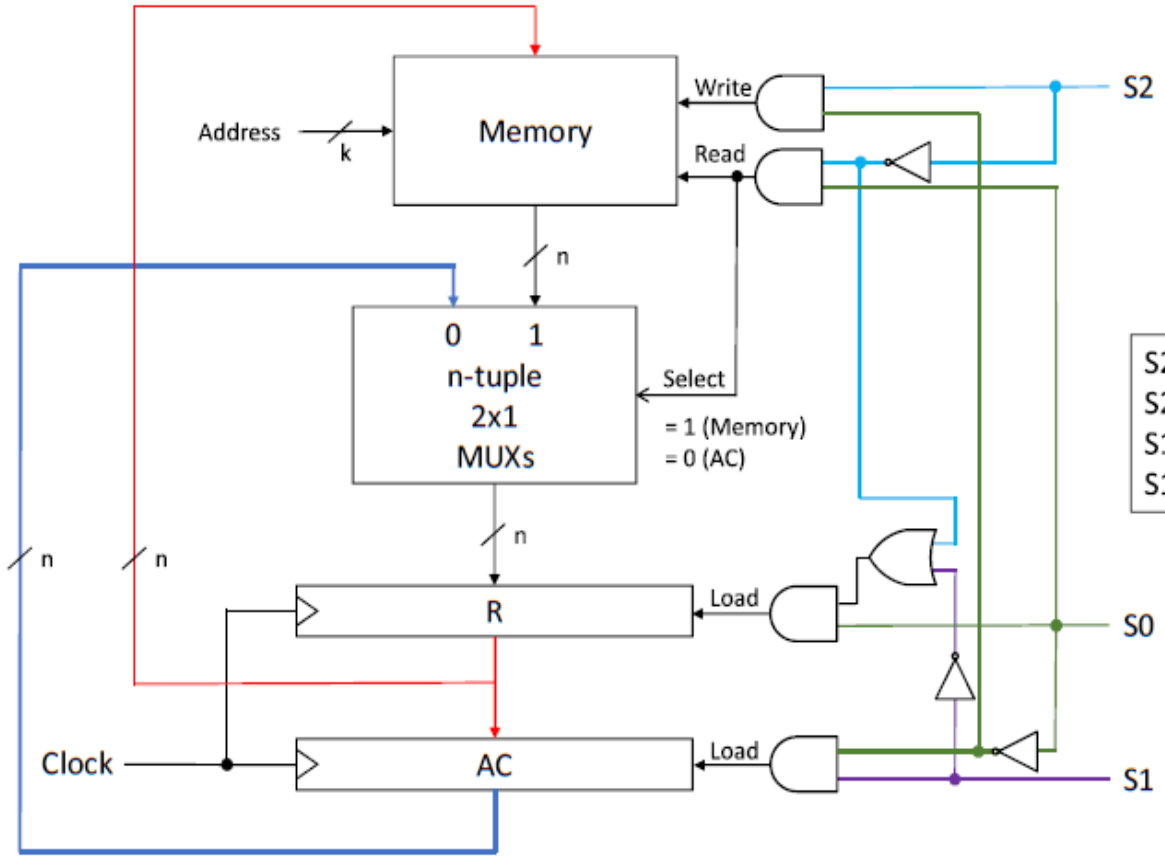
Question 3

- The register transfer statements for read and write operations (from and to a memory via a register R) and data transfers between AC and R are given in the table.
- Words in the ***memory are n bits long*** and the memory comprises of 2k words.
- ***Draw the block diagram for hardware implementation***.
- Show the control functions to select each block in the diagram.
- The block diagram should include all details (all relevant connections and labels) unambiguously so that a technician can implement it without asking any questions. Submission includes a clearly drawn block diagram.

Control Signal	Statement	Operation
S2' S0	R ← M[AR]	Memory Read
S2 S0'	M[AR] ← R	Memory Write
S1' S0	R ← AC	Copy from accumulator to register
S1 S0'	AC ← R	Copy from register to accumulator

Solution 3

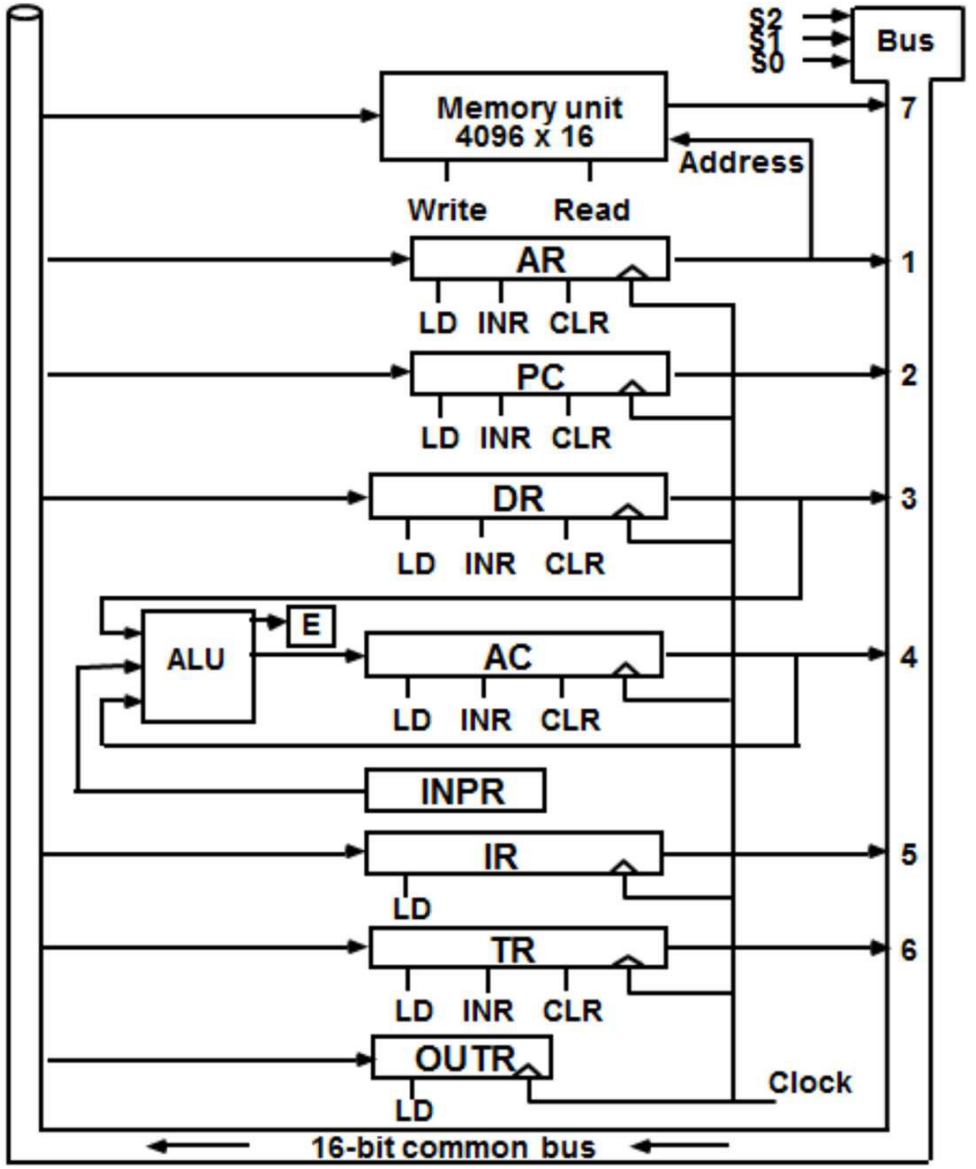
Control Signal	Statement	Operation
S2' S0	R ← M[AR]	Memory Read
S2 S0'	M[AR] ← R	Memory Write
S1' S0	R ← AC	Copy from accumulator to register
S1 S0'	AC ← R	Copy from register to accumulator



S2'S0 : Memory Read
S2 S0': Memory Write
S1'S0 : R ← AC
S1 S0': AC ← R

Question 4

- Direct addressing mode (I = 0) and indirect addressing mode (I = 1) of the Basic Computer will be changed with PC-relative addressing.
  - In direct PC-relative addressing (I = 0) the effective address (EA) is computed as:
    - EA = PC + TR(11-0)
  - In indirect PC-relative addressing (I = 1) the effective address (EA) is computed as:
    - EA = M[PC + TR(11-0)]
- Implement fetch, decode and execute cycles of LDA instruction (AC←M[EA], opcode=010) with direct and indirect register-indexed addressing mode using RTL and timing signals T0, T1, etc.
- For this question, use the simple computer architecture given in the figure below. Ignore interrupt signal.



Solution 4

Fetch and Decode [Store Address in TR]:

T0: AR ← PC ($S_0S_1S_2$=010, T0=1)
T1: IR ← M [AR], PC ← PC + 1 (S0S1S2=111, T1=1)
T2: D0, . . . , D7 ← Decode IR(12-14), AR ← IR(0-11), I ← IR(15)

Set AR ← PC+TR to compute EA, do the following BOTH in Direct (I=0) and Indirect Relative Addressing (I=1), we don't need to check I bit

D7'T3: DR ← PC
D7'T4: AC ←DR, DR ← TR
D7'T5: AC ← AC + DR
D7'T6: AR ←AC

InDirect Relative Addressing:

D7'IT7: AR ← M[**AR**] [I=1 is INDIRECT-RELATIVE ADDRESSING]

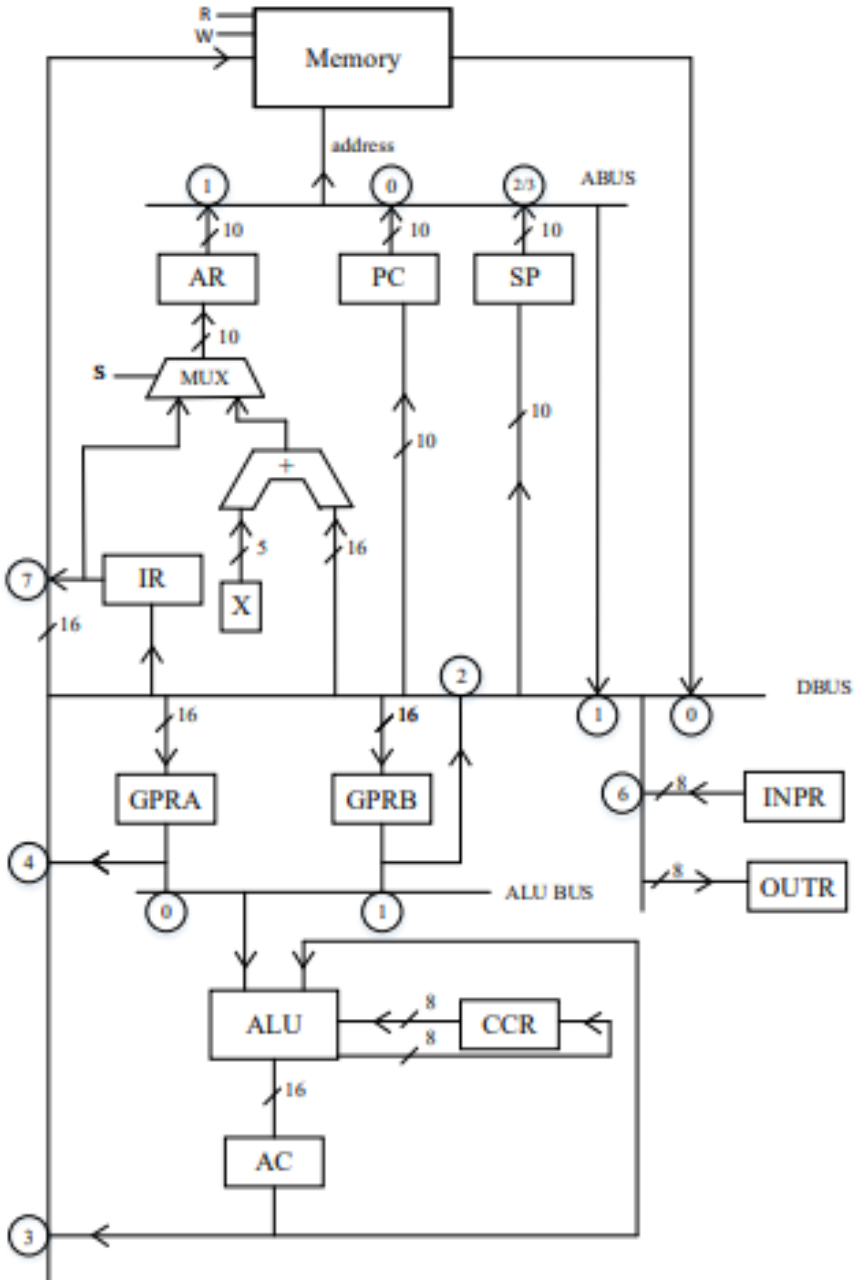
Direct Relative Addressing:

D7'I'T7: Nothing [I=0 is DIRECT-RELATIVE ADDRESSING]

LDA (Load to AC) [NO CHANGE]:

D2T8: DR ← M[**AR**]
D2T9: AC ← DR, SC ← 0

Question 5



Control signal			Register to write DBUS
D_2	D_1	D_0	
0	0	0	Memory
0	0	1	ABUS
0	1	0	GPRB
0	1	1	AC
1	0	0	GPRA
1	0	1	X
1	1	0	INPR
1	1	1	IR

S	MUX Output
0	IR(9-0)
1	X+DBUS(9-0)

Control signal		Register to write ABUS
A_1	A_0	
0	0	PC
0	1	AR
1	dont care	SP

Control signal M	Register to write ALUBUS
0	GPRA
1	GPRB

The memory has R/W control signals. The instruction format is as follows:

15 0

2-bit	4-bit	10-bit
addr. mode	opcode	operand address

There are 4 different addressing modes:

Addr. mode		Effective address (EA)	mode
0	0	$EA \leftarrow IR(9-0)$	direct
0	1	$EA \leftarrow M[IR(9-0)]$	indirect
1	0	$EA \leftarrow IR(9-0)+X$	indexed
1	1	$EA \leftarrow SP$	stack

Inside the control unit, the opcode bits of the IR register is connected to an opcode decoder that generates signals from K_0 to K_{15}. In addition there are two flags (R_1 and R_0) that are connected to the address mode bits in IR: $R_1 \leftarrow IR(15)$ and $R_0 \leftarrow IR(14)$. Finally, there is a 4-bit sequence counter in the control unit that is connected to a decode that generates timing signals of $T_0, T_1, ..., T_{15}$.

Solution 5.a

❑Write the microoperations at fetch & decode cycles in register transfer language (RTL) and the control signals.

Solution :

- $\boldsymbol{T_0}$: $\boldsymbol{IR \leftarrow M[PC], PC \leftarrow PC + 1, X \leftarrow 0}$
  - $\boldsymbol{(A_1A_0 = 00\ , R = 1, D_2D_1D_0 = 000, IR_{LD} = 1)}$
- $\boldsymbol{T_1}$: $\boldsymbol{K_0 \ldots K_{15} \leftarrow Decode\ IR(13-10), AR \leftarrow X + IR(9-0), R_1R_0 \leftarrow IR(15-14)}$
  - $\boldsymbol{(D_2D_1D_0 = 111, AR_{LD} = 1, S = 1)}$

Control signal		Register to write ABUS
A_1	A_0	
0	0	PC
0	1	AR
1	dont care	SP

Control signal			Register to write DBUS
D_2	D_1	D_0	
0	0	0	Memory
0	0	1	ABUS
0	1	0	GPRB
0	1	1	AC
1	0	0	GPRA
1	0	1	X
1	1	0	INPR
1	1	1	IR

The memory has R/W control signals. The instruction format is as follows:

15 0

2-bit	4-bit	10-bit
addr. mode	opcode	operand address

Solution 5.b

- Consider an instruction named XCH (opcode=9) that exchanges the values of GPRA and GPRB without data loss.
- **XCH: GPRA←GPRB, GPRB←GPRA**

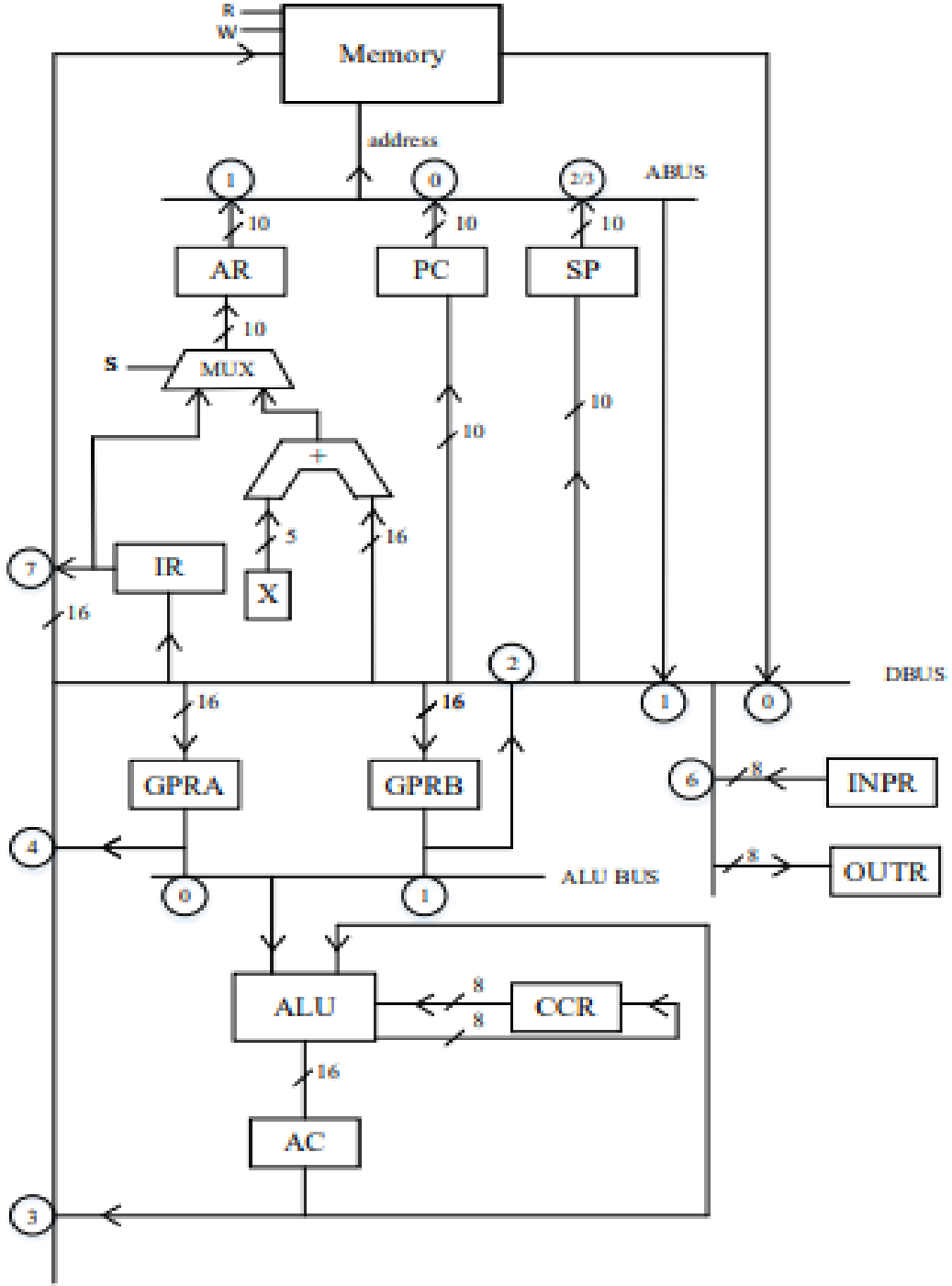
Write the microoperations that execute XCH instruction in RTL. and mention the corresponding control signals at each clock cycle.

Control signal M	Register to write ALUBUS
0	GPRA
1	GPRB

Control signal			Register to write DBUS
D_2	D_1	D_0	
0	0	0	Memory
0	0	1	ABUS
0	1	0	GPRB
0	1	1	AC
1	0	0	GPRA
1	0	1	X
1	1	0	INPR
1	1	1	IR

Solution :

- $\boldsymbol{T_2K_9}$: $\boldsymbol{AC \leftarrow GPRA(transfer), GPRA \leftarrow GPRB}$
  - $\boldsymbol{(M = 0, D_2D_1D_0 = 010, GPRA_{LD} = 1, AC_{LD} = 1)}$
- $\boldsymbol{T_3K_9}$: $\boldsymbol{GPRB \leftarrow AC}$
  - $\boldsymbol{(D_2D_1D_0 = 011, GPRB_{LD} = 1)}$



Question 6

- Consider the instruction MAD (memory add, opcode=3) that adds the data in two consecutive effective addresses (EA) in memory, and write the result to the next effective address in memory.
- **MAD: M[EA+2] ← M[EA]+M[EA+1]**
- Note: Your answers should be optimum in terms of the required clock cycles. You can ignore ALU control signals. You can overwrite values of the registers.

Solution 6.a

- Write the address decoding cycle in RTL, and mention the corresponding control signals. It should be performed in one clock cycle.
- **Solution :**
- $T_2K_3R_0'R_1'$: **DO NOTHING**
- $T_2K_3R_0R_1'$: $AR \leftarrow X + M[AR]$
  - $(A_1A_0 = 01,\ R = 1, D_2D_1D_0 = 000, S = 1)$
- $T_2K_3R_0'R_1$: $AR \leftarrow X + AR$
  - $(A_1A_0 = 01, D_2D_1D_0 = 001, S = 1)$
- $T_2K_3R_0R_1$: $AR \leftarrow SP$
  - $(A_1 = 1, D_2D_1D_0 = 001, S = 1)$

S	MUX Output
0	IR(9-0)
1	X+DBUS(9-0)

There are 4 different addressing modes:

Addr.	mode	Effective address (EA)	mode
0	0	$EA \leftarrow IR(9-0)$	direct
0	1	$EA \leftarrow M[IR(9-0)]$	indirect
1	0	$EA \leftarrow IR(9-0) + X$	indexed
1	1	$EA \leftarrow SP$	stack

Control signal			Register to write DBUS
D_2	D_1	D_0	
0	0	0	Memory
0	0	1	ABUS
0	1	0	GPRB
0	1	1	AC
1	0	0	GPRA
1	0	1	X
1	1	0	INPR
1	1	1	IR

Control signal		Register to write ABUS
A_1	A_0	
0	0	PC
0	1	AR
1	dont care	SP

Solution 6.b

- Write the microoperations that execute **MAD instruction in RTL** (after the address decoding cycle), and mention the corresponding control signals (that should be 1) at each clock cycle.
- **MAD: M[EA+2] ← M[EA]+M[EA+1]**
- **Solution :**
- $\boldsymbol{T_3K_3}$: $\boldsymbol{GPRA \leftarrow M[AR], AR \leftarrow AR + 1}$
  - $\boldsymbol{(A_1A_0 = 01\,, D_2D_1D_0 = 000, GPRA_{LD} = 1, AR_{INC} = 1, R = 1)}$
- $\boldsymbol{T_4K_3}$: $\boldsymbol{GPRB \leftarrow M[AR], AR \leftarrow AR + 1, AC \leftarrow GPRA(transfer)}$
  - $\boldsymbol{(A_1A_0 = 01, D_2D_1D_0 = 000, GPRB_{LD} = 1, AC_{LD} = 1, M = 0, R = 1)}$
- $\boldsymbol{T_5K_3}$: $\boldsymbol{AC \leftarrow AC + GPRB}$
  - $\boldsymbol{(M = 1, AC_{LD} = 1, GPRB_{LD} = 0)}$
- $\boldsymbol{T_6K_3}$: $\boldsymbol{M[AR] \leftarrow AC}$
  - $\boldsymbol{(D_2D_1D_0 = 011, W = 1, A_1A_0 = 01, AC_{LD} = 0)}$

Control signal			Register to write DBUS
D_2	D_1	D_0	
0	0	0	Memory
0	0	1	ABUS
0	1	0	GPRB
0	1	1	AC
1	0	0	GPRA
1	0	1	X
1	1	0	INPR
1	1	1	IR

Control signal		Register to write ABUS
A_1	A_0	
0	0	PC
0	1	AR
1	dont care	SP