UNIVERSITE DE BOUMERDES UNIVERSITY OF BOUMERDES	***IGEE*** ***University M'Hamed*** ***Bougara of Boumerdès***	***Last Name:*** …… ***First Name:*** …… ***Group:*** ……	***Grade :***

EE321: Computer Architecture
Control Solution (1h30)
2016 – 2017

Notes: Answer briefly and clearly using the provided space. No extra sheet will be accepted.

1. Simple CPU Function and Structure (14 pts)

Consider the simple 12-bit CPU given in Figure 1 with the following specifications:

- 12-bit address bus.
- The instruction format provides 4 bits for the opcode and 8 bits for the operands.

1.1 Give the size of data registers, IR, PC, and memory address register. Justify

Data registers: 12 bits (12-bit CPU)
IR: 12 bits (4 bits for opcode and 8 bits for operands)
PC and MAR: 12 bits (12-bit address bus)

1.2 How many total registers does our simple CPU have? Justify.

R0 to R9: 10 registers

1.3 What is the maximum addressable locations? Justify.

Since we have 12-bit address bus, the maximum addressable locations is $2^{12} = 4096\ locations = 4\ K\ locations.$

1.4 What is the maximum addressable memory size? Justify

Since we have 12-bit data bus, the maximum addressable memory is $12 \times 2^{12} = 49\ 152\ bits = 48\ Kbits = 6\ KBytes.$

1.5 Does our simple CPU have accumulator? Justify.

Yes, R7 register used as one of the ALU input and to save ALU result.

1.6 Which signal is the intenal data bus? What is its size?

Output signal of the 8-to-1 multiplexer. Internal data bus size is 12 bits.

1.7 Which of the registers R0 to R9 is a) program counter, b) memory buffer register, c) memory address register, and d) instruction registser? Justify.

a) PC = R5, connected to the increment circuit.
b) MBR = R6 , connected to the data bus.
c) MAR = R8, connected to the address bus.
d) IR = R9, it is the input of the control unit.

1.8 What is the function of the increment circuit?

It increments PC.

1.9 Based on the simple CPU architecture, which signals are generated by the control unit?

C0 … C9 to enable each of the 10 registers.
S7 … S0 to control the ALU operation, the data being written to the internal data bus, and the data being written to MBR and PC.

1.10 Based on the simple CPU architecture, give the micro-operations and time units for fetch cycle.

t1: R8 <-- R5 (MAR<--PC)
t2: R6 <-- data in (MBR<--data in)
t3: R9 <-- R6 (IR<-- MBR)
R5 <-- R5+I (PC<--PC+I)

1.11 Give the active control signals for each micro-operation of fetch cycle.

t1: R8 <-- R5 ==> active signals are: C8, S3 and S5
t2: R6 <-- data in ==> active signals are: C6 and S6
t3: R9 <-- R6 ==> active signals are: C9, S4 and S5
R5 <-- R5+I ==> active signal is: C5

1.12 Give the microoperations and time units for execute cycle of **MV Rd,Rs** instruction given in Table 1.

t1: Rd <-- Rs

1.13 Give the microoperations and time units for execute cycle of **ADR Rd,Rs** instruction.

t1: R7 <-- Rd (Acc<--Rd)
t2: R7 <-- R7 + Rs (Acc<--Acc+Rs)
t3: Rd<-- R7 (Rd<--Acc)

1.14 Give the microoperations and time units for execute cycle of **STR Rd,Rs** instruction.

t1: R8 <-- Rd (MAR<--Rd)
t2: R6 <-- Rs (MBR<--Rs)
t3: data out <-- R6 (data out <--MBR)

2. Interrupt cycle (6 pts)

Consider the micro-operations of the interrupt cycle given in Table 2.

2.1 What is save_address?

Address to use to save the address of the next instruction (content of PC)

2.2 What is ISR_address?

Address of the first instruction of interrupt routine.

2.3 Is-it possible to combine micro-operations 1 and 2 in the same clock cycle? Justify.

Yes if MAR, MBR, and PC are not connected by the same internal data bus.

2.4 Is-it possible to combine micro-operations 3 and 4 in the same clock cycle? Justify.

Yes, there is no conflict and the two operations do not use the same data bus.

2.5 Consider a maskable interrupt of Z80 processor and the following data and register values: A=00H, F=10H, PC=F00H, IR=0F00, SP=00F0H, I= 0FH, R=F0H, and data bus=03H. Give the value of save_address and ISR_address in case of a) interrupt mode 1, and b) interrupt mode 2.

a) Save_address = SP = 00F0H, and ISR_address = 0038H
*b) Save_address = SP = 00F0H, and ISR_address = I*256 + data bus value = 0F03H*

Good Luck!

Instruction	Opcode	Description
LDI Rd, X	1H	Rd = X
LDD Rd, X	2H	Rd = [X]
LDP Rd, X	3H	Rd = [R0+X]
LDR Rd, Rs	4H	Rd = [Rs]
STR Rd, Rs	5H	[Rd] = Rs
ADR Rd, Rs	6H	Rd = Rd+Rs
ADM Rd,X	7H	Rd = Rd+X
ADD Rd, X	8H	Rd =Rd+ [X]
ADN Rd,X	9H	Rd=Rd+[[X]]
ADP Rd,X	AH	Rd=Rd+[R0+X]
MV Rd, Rs	BH	Rd = Rs
HLT	FH	Stop the execution of the program

Rd = Destination register (it can be any GPR).
Rs = Source register (it can be any GPR)
R0 = Register R0.
[X] = content of memory location X.

Table 1: Instructions description.

Clock cycle	MOP Nb.	Micro-operation (MOP)
T1	*1*	*MBR ← PC*
T2	*2* *3*	*MAR ← save_address* *PC ← ISR_address*
T3	*4*	*Memory ← MBR*

Table 2: Micro-opérations for interrupt cycle.

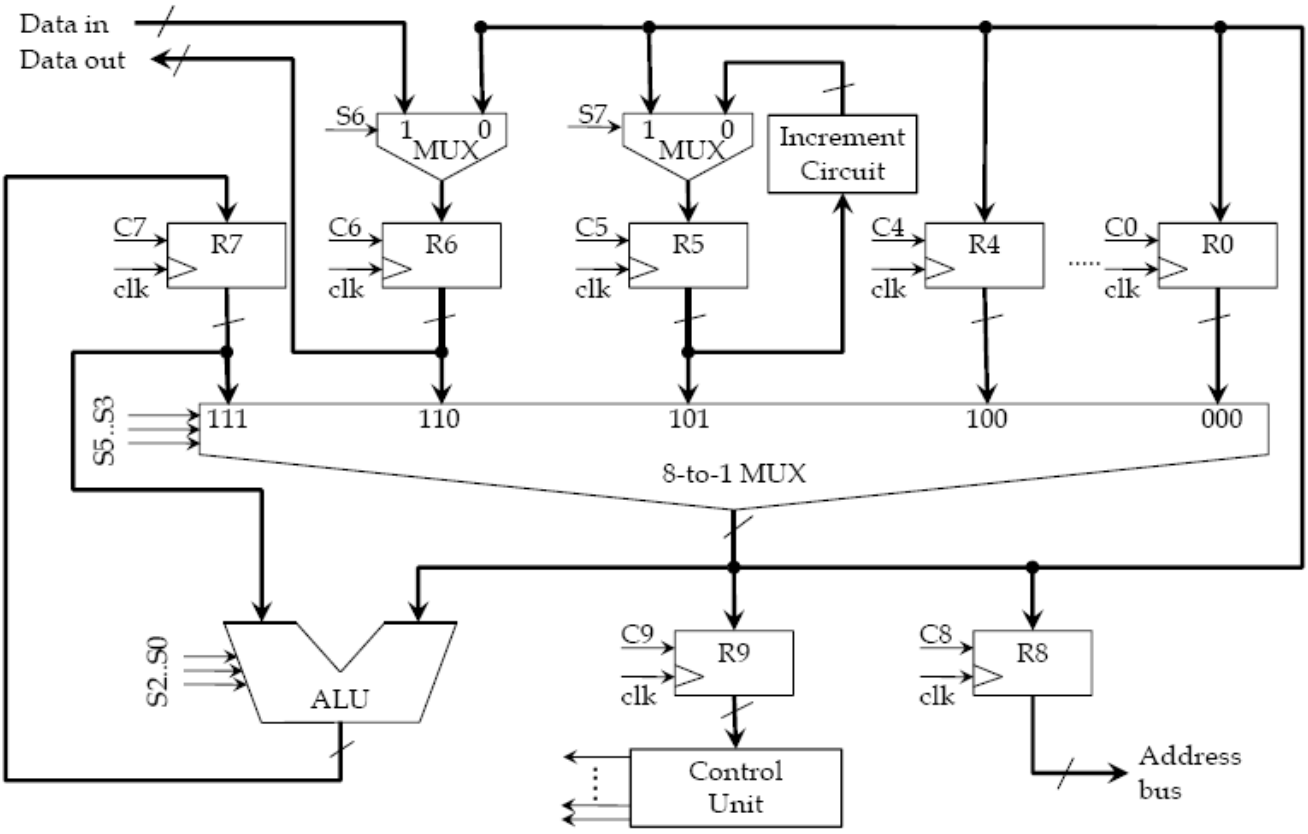

Figure 1: Simple CPU architecture.

Register	R0	R1	R2	R3	R4	R5	R6	R7	R8	R9	SP
Code	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010

Table 3: Binary codes of GPR.

11	10	9	8	7	6	5	4	3	2	1	0
Opcode				Destination register				Source operand			

Table 4: Data processing instruction format.