IGEE University M'Hamed Bougara of Boumerdès	Last Name: .... First Name: .... Group:	Grade :

EE323: Microprocessor Systems Design
Control (1h30)
2019 – 2020

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

1. Z80 Assembly Program Execution (7 pts)

Consider the diagnostic routine program given in Table 1.

1.1 Give the size, the number of M-cycles, and the number of T-states of the program.

1.2 Explain the HEX code of `JR START` instruction.

1.3 How many fetch cycles, memory read cycles, and memory write cycles are executed in one loop?

1.4 How many times Z80 inserts a) $\overline{WR}$, b) $\overline{RD}$, c) $\overline{M1}$, and d) $\overline{MREQ}$ signals in one loop?

1.5 Assuming 4 MHz clock frequency, give the time interval between two $\overline{WR}$ pulses.

1.6 Give the HEX value on the data bus and address bus when Z80 inserts $\overline{WR}$ signal.

1.7 Give the state (low or high) of the $\overline{M1}$, $\overline{MREQ}$ and $\overline{RD}$ signals when Z80 inserts $\overline{WR}$ signals.

2. Z80 Assembly Programming Language (7 pts)

2.1 List the differtent ways to specify an operand for Z80 instruction.

2.2 List the differtent ways to specify a) source, and b) destination operand for Z80 8-bit arithmetic instructions.

2.3 Write a simple assembly program to multiply two 8-bit unsigned numbers stored in memory locations 3000H and 3001H (Hint: $N * M = \sum_{i=1}^{M} N$). The low byte of the result should be stored at address 3002H and the high byte at address 3003H.

No	Instruction	No	Instruction
1		9	
2		10	
3		11	
4		12	
5		13	
6		14	
7		15	
8		16	

2.4 A faster algorithm to multiply two 8-bit unsigned numbers is given in Figure 1. Modify the assemby program of the previous question to implement this algorithm.

No	Instruction	No	Instruction
1		*12*	
2		*13*	
3		*14*	
4		*15*	
5		*16*	
6		*17*	
7		*18*	
8		*19*	
9		*20*	
10		*21*	
11		*22*	

3. Memory Interfacing (6 pts)

Consider a memory interfacing circuit that connect Z80 to an SRAM memory chip with starting address 9400H. The memory map is 9400H-94FFH and the foldback memory ranges are: 9500-95FFH, 9600H-96FFH, and 9700H-97FFH.

3.1 What is the depth and the size of the interfaced memory? Justify.

3.2 What is the total used memory space? Justify.

3.3 Give the Z80 addresses that can be used to point the memory location 23H.

3.4 Give the "don't care" lines in the interfacig circuit.

3.5 Draw the interfacing circuit using 3-to-8 decoder.

Good Luck!

No	Instruction	HEX code	T-states
1	`START: LD A, 0F7H`	3EF7H	7 (4,3)
2	`LD (2800H), A`	320028H	13 (4,3,3,3)
3	*`JR START`*	*18F9H*	*12 (4,3,5)*

Table 1: Diagnostic routine.

To multiply two 8-bit unsigned numbers N and M, we can use the following equation:

$$N * M = \sum_{i=0}^{7} \left(n_i * 2^i * M\right), \quad n_i \text{ is the bit number } i \text{ of } N$$

$$N * M = n_0 * M + 2n_1M + 2^2n_2M + \cdots + 2^6n_6M + 2^7n_7M$$

$$N * M = n_0 * M + 2(n_1M + 2(n_2M \ldots + 2(n_6M + 2(n_7M))))))$$

The corresponding pseudocode is:

```
N and M are the two 8-bit unsigned numbers
LP is the low byte of the product (result)
HP is the high byte of the product (result)

BEGIN
     Initialize LP to zero
     Initialize HP to zero
     FOR i=0 to 7
          LP = LP + LP                (i.e. LP = 2*LP)
          Rotate left HP              (i.e. HP = 2*HP + carry)
          Shift left arithmetic N     (i.e. get ni in carry flag)
          IF (carry = 1)              (i.e. if ni = 1)
               LP = LP + M
               IF (carry =1)          (i.e. if overflow)
                    Increment HP
END
```

Figure 1: Pseudocode for multiplication of two 8-bit unsigned numbers.

Instruction	Description
`RLC r`	The bits of register r are shifted left. The bit 7 is copied to the carry flag and to bit 0. The other bits are shifted left.
`RL r`	The bits are shifted left. The carry is copied to bit 0 and the bit 7 is copied to the carry flag.
`RRC r`	The bits are shifted right. The bit 0 is copied to the carry flag and to bit 7.
`RR r`	The bits are shifted right. The carry is copied to bit 7 and the bit 0 is copied to the carry flag.
`SLA r`	The bits are shifted left. The content of bit 7 is copied to the carry flag and '0' to bit 0.
`SRA r`	The bits are shifted right. The bit 0 is copied to the carry flag and bit 7 remains unchanged.
`SRL r`	The bits are shifted right. The bit 0 is copied to the carry flag and '0' to bit 7.

Table 2: Z80 shift and rotate instructions.