Memorial University of Newfoundland

Engineering 4862 MICROPROCESSORS

Assignment 1 Solution

***0**. What is pipelining? Is it possible to have more than two units working in parallel in a microprocessor? If yes, give an example of such a microprocessor. If no, explain why not.*

Pipelining is when two or more units work together simultaneously on different stages of different instructions. These different units are all located within the CPU. The functioning is similar to an assembly line: when one unit inside the CPU finishes processing its aspect of an instruction, the next unit takes over.

In the 8086/8088, the bus interface unit (BIU) fetches instruction bytes and loads them into the queue while the execution unit (EU) performs an action according to a previously loaded instruction. Overall, this is more efficient in terms of resource usage than a non-pipelined system.

It is possible for a microprocessor to have many more units in a pipeline. For example, the 80386 processor has 4 units that operate in parallel. The Pentium microprocessor has two separate pipelines with 5 units (stages) in each, for a total of 10 units.

***1**. Convert the following to binary, decimal, and hexadecimal:*

	Bin	Dec	Hex
a. 50_{10}	110010	50	32
b. 101111011_2	101111011	379	17B
c. 512_{10}	1000000000	512	200
d. $E29_{16}$	111000101001	3625	E29
e. 123_{10}	1111011	123	7B
f. $1FFF_{16}$	1111111111111	8191	1FFF

***2**. Perform Addition and Subtraction for the following HEX numbers*

a. 24FH-129H	=	126H		
b. F94H+5C6H	=	155AH	or	55AH & CF=1
c. FFFFH+1111H	=	11110H	or	1110H & CF=1
d. 2FFFFH-FFFFFH	=	30000H & BF/CF =1		
e. EF9H-5BCH	=	93DH		
f. 5FC54H-3DFE5H	=	21C6FH		

***3**. Convert -32_{10} and 71_{16} to binary, and add the numbers together. Can the original and final numbers be stored in 8-bit registers?*

32 = 00100000
2's complement of 32 = 1101 1111 + 1 = 1110 0000
71 = 0111 0001
Summary 0101 0001 & CF =1
CF is neglected for signed number and the remaining 8-bit hold the result.

***4**. Explain how the 8086/8088 uses 16-bit segment registers to access memory in a 1MB address space.*

The 8086/8088 are 16-bit microprocessors, and thus all internal registers are 16-bits large. In addition, operands for instructions can at most be 16-bit numbers. Thus the 8086/8088 must somehow create a 20-bit number from these 16-bit numbers.

The solution is to use several registers as segment registers. The numbers in these registers are multiplied by 16_{10} (10H, or 10000B) to create a new value that is 20 bits long. The value is a base address for a segment, and can be used to specify a memory location anywhere within a 1MB address space (note that to address 1MB, 20 address lines are necessary). However, the generated addresses only refer to those memory addresses that are multiples of 10H. To refer to every possible location, and offset must be added to the segment's base. For the 8086/8088, the offset is a 16-bit number as well, which means that 64KB locations can be addressed if the segment register remains constant.

Thus the physical address used to access memory is calculated by

(segment register) x 10H + offset = address

***5**. Calculate the physical addresses from the following logical addresses and give the lower and upper range of that segment:*

	Physical address	Lower range	Upper range
a. 100EH:34D2H	135B2H	100E0H	200DFH
b. EF01H:0001H	EF011H	EF010H	FF00FH
c. 010AH:2EDBH	03F7BH	010A0H	1109F

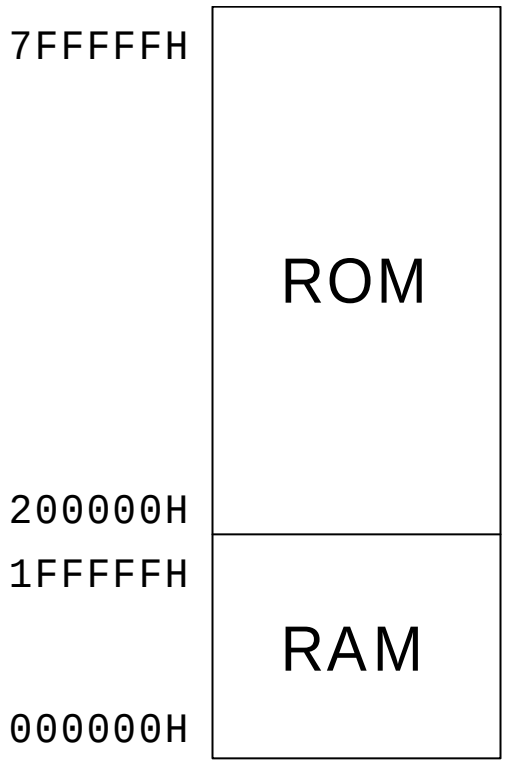
***6**. Calculate 3 different logical addresses from the physical address 4E276H.*

1. 4E27:0006
2. 4E20:0076
3. 4E00:0276

***7**. Suppose that for a certain device to operate, it requires 2MB of RAM and 6MB of ROM. What is the minimum number of address lines the microprocessor must support? Draw a memory map of the system; assume that the addresses for the RAM are below the addresses for the ROM.*

For the minimum number of address lines, we must assume that all the memory is packed as tightly together in the memory map as is possible.
Thus the total amount of possible memory for the system is 2MB + 6MB = 8 MB. 8MB of memory is 8 x 2^{20} or 2^{23} bytes. This means that 23 address lines are required.

The memory map is shown on the side.



***8**. In a given byte-addressable computer, memory locations 10000H to 9FFFFH are available for user programs. The first location is 10000H and the last location is 9FFFFH. Calculate the following:*
a. The total number of bytes available (in decimal)
b. The total number of kilobytes (in decimal)

a. 9FFFF – 10000 + 1 = 90000H = 589824 Byte
b. 1 KB = 1024 Bytes, thus available 576 KB.

***9**. Identify the addressing mode used in each of the following instructions, and state the size of the operands:*

a. MOV DX, [47H]	Direct addressing mode	16-bit
b. MOV [BP+0DH], DH	Based Relative AM	8-bit
c. ADD AX, BP	Register AM	16-bit
d. MOV BX, 8H	Immediate AM	16-bit
e. INC AX	Register AM	16-bit

10. CPU A has a 8-bit data bus and 16-bit address bus, CPU B has a 16-bit data bus and 32-bit address bus. Give the maximum value that can be brought into the CPU at a time (in HEX and Decimal) and the maximum addressable memory for these two CPUs.

***11**. Which of the following instructions cannot be coded in 8088/86 Assembly language? Give the reason why not, if any. All numbers are in HEX*

a. MOV AX, 27	*b. MOV AL, 97F*	*c. MOV DS, 9BF2*
d. MOV CX, 397	*e. MOV SI, 9516*	*f. MOV CS, 3490*
g. MOV DS, AX	*h. MOV BX, CS*	*i. MOV CH, AX*

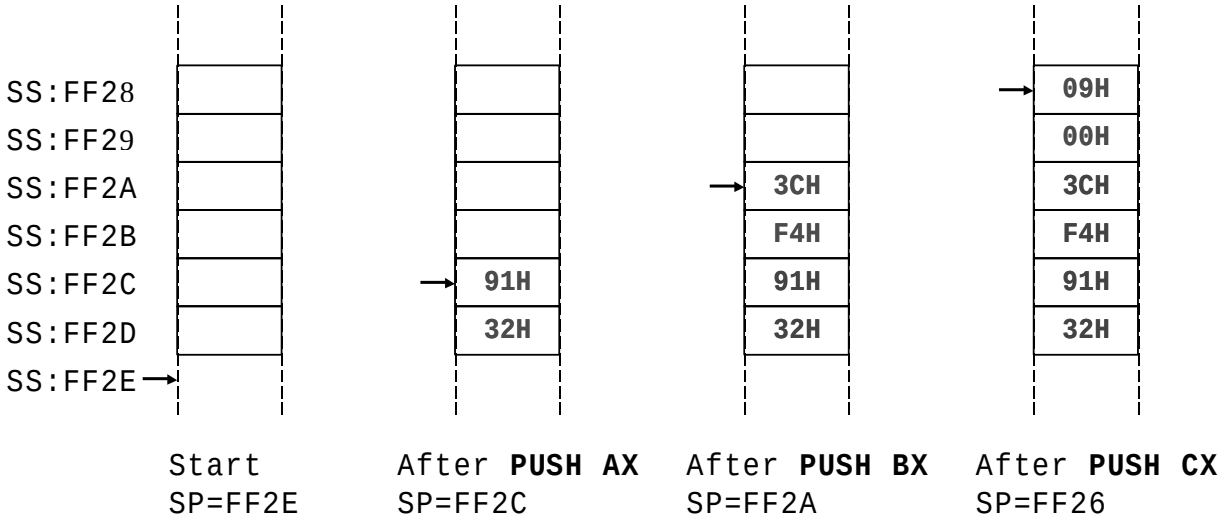
j. MOV AX, 23FB9 *k. MOV CS, BH* *l. MOV AX, DL*

Error Ones:

b.	97F exceed AL can hold
c.	can't move immediate number to segment register
f	same as c
i	source and destination register should match
j	23FB9 exceed AX range
k	same as i
l	same as i

12*. Assume that SP=FF2EH, AX=3291H, BX=F43CH and CX=09. Find the content of the stack and stack pointer after the execution of each of the following instructions*

PUSH AX
PUSH BX



13. *Find the status of the CF, PF, AF, ZF and SF for the following operations*

a.	*b.*	*c.*
MOV BL, 9FH	*MOV AL, 23H*	*MOV DX, 10FFH*
ADD BL, 61H	*ADD AL, 97H*	*ADD DX, 1*

A. CF=1, PF=1, AF=1, ZF=1, SF=0
B. CF=0, PF=0, AF=0, ZF=0, SF=1
C. CF=0, PF=1, AF=1, ZF=0, SF=0

14*. Assume that the registers have the following values in HEX and that CS=1000, DS=2000, SS=3000, SI=4000, DI=5000, BX=6080, BP=7000, AX=25FF, CX=8791 and DX=1299. Calculate the physical address of the memory where the operand is stored and the contents of the memory locations in each of the following addressing examples:*

a. MOV [DI][BX]+28, CX *b. MOV [SI+BX+8], AH* *c. MOV [BX], AX*
d. MOV [BP+SI+100], BX *e. MOV [DI+BP+100], AX*

	Physical address	Content
A.	2B0A8H	91
	2B0A9H	87
B.	2A088H	25
C.	26080H	FF
	26081H	25
D.	3B100H	80
	3B101H	60
E.	3C100H	FF
	3C101H	25