EEET2505_Mock Assessment 1_2023Cb

Friday, 17 November 2023 8:25 am

EEET2505... Theory...

EEET2505 – Introduction to Embedded Systems

Mock Assessment 1

Student Name_____Student ID_____

Total Mark:

Important Notes:

1. This is an open book assessment. You can use your note or Internet resources during the session. You are not allowed to chat/communicate with other during the assessment.
2. Write your name (First Name and Last Name on top of this paper)
3. Please give the details of your solutions in the spaces provided on the paper itself. For the mock assessment, save the file and upload to Canvas. For the actual assessment, you will submit the paper after completion.
4. You are recommended to attempt all of questions provided.
5. Your solution should be detailed.
6. If you have any questions, please be in contact with the lecturer for further advice.

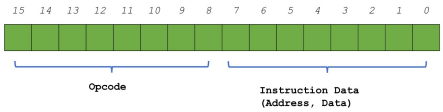
A Microcontroller (MCU) is given with:

- An Accumulator **A** (8 bit)
- Program Counter (**PC**)

The initial instruction set is listed in the following table:

No	Instruction	Mnemonic code	Opcode	Description
1	No Operation	NOP	0000 0000 0x00	No effect on internal registers.
2	Load data	LDA data For example LAD 0x05	0000 0001 0x01	Load the accumulator with data. **A←data**
2	Store Accumulator to memory	STA addr For example STA 0x01	0000 0010 0x02	Store the contents of the accumulator in memory at address addr. **mem[addr] ← A**
3	Add Accumulator with a data from memory	ADD addr For example ADD 0x03	0000 0011 0x03	Add the contents at address addr to the contents of the accumulator. **(A) ← (A) + mem[addr]** *Note – if the Sum is overflow (over 8 bit), a Carry flag C will be set to 1.*
4	Jump to Address	JMP addr For example JMP 0x01	0000 0100 0x04	Jump to address addr. **(PC) ← addr**

The generic instruction (16-bit) has the format as follows:



Couple notes for the instructions:

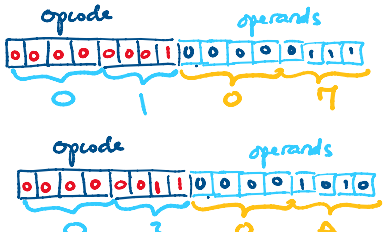
- The CPU work with data that is 8 bit long (or 2 Hexadecimal digit).
- The address is 8-bit long.
- For instructions where we don't need operands or data instructions, the remaining least significant bits of the instruction will be Zeros.
- For each instruction, after the instruction is fetched from the memory, the **Program Counter** (PC) will increase by 1 to point to the next instruction, except for the instruction **JC,** which will make the Program Counter will point to a designated address.

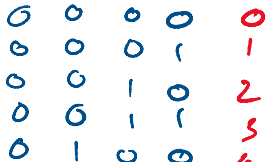
1.1 – Program

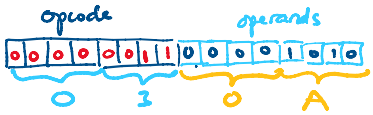
You are given with a simple program as follows. Do:

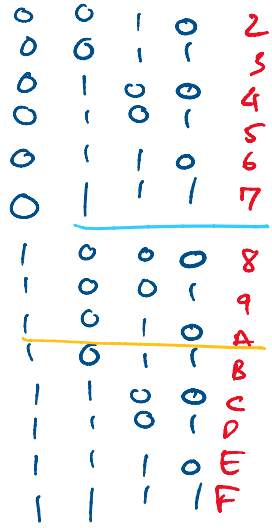
- Write the corresponding mnemonic code for each instruction and then translate the instruction into corresponding machine code. An example is done for your information.
- Indicate the PC for the instruction that will be executed next.
- Determine the value of Accumulator once the program is over.

Assume that initially PC = 0x00, Accumulator are reset to Zero. *Type your answers in the designated space highlighted in Orange).*









Address	Tasks	Mnemonic code (Assembly code)	Machine code (in HEX)	Program Counter for **Next** instruction (In Binary)
0x00	Load Accumulator with the data 0x07	LDA 0x07	0x 0x0107	
0x01	Add the accumulator with the content at the address 0x0A	ADD 0x0A	0x 0x030A	
0x02	No Operation	NOP	0x0000	
0x03	Store the value of Accumulator to the memory location 0x10	STA 0x10	0x 0x0210	
0x04	Jump to the Address 0x06	JMP 0x06	0x 0x0406	0b0000000110
0x05	Add the Accumulator with the content at the address 0x11	ADD 0x11	0x 0x0311	
0x06	No Operation	NOP	0x 0x0000	
0x07	Store the value of Accumulator to the memory location 0x12	STA 0x12	0x 0x0212	
.....				
.....				

ACC

Initially ACC = 0

ACC = 0x07

ACC = 0x07 + 0x09 = 0x10

ACC = 0x10

→ Addresses on the Memory

→ pre-loaded onto the memory

0x0A	0x09	
.....		
0x10	0x	0x10
0x11	0x01	
0x12	0x	0x10
End of program		

Accumulator at the end of the program in HEX

Acc = 0x 0x10

0x09 = 0b 0000 0111

+ 0x09 = 0b 0000 1001

0b 0001 0000

0x 1 0