Liam Spinner

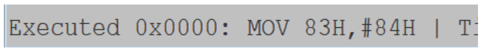
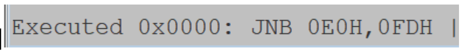
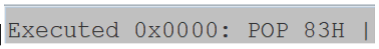
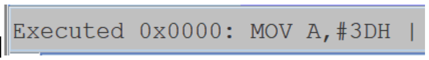
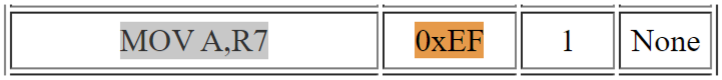
CSE2120

Dr. Caraway

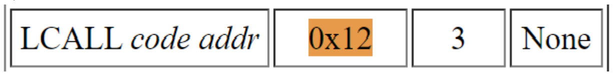
30 Mar 2024

Homework 5

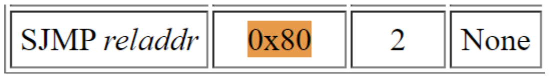
Chapter 3: 3.2, 3.4, 3.9, 3.12, 3.29 on pages 78-81 (93-95 of pdf)

- 3.2 What are the hexadecimal bytes for the following instructions?
  - https://www.win.tue.nl/~aeb/comp/8051/set8051.html#51mov
  - a. MOV DPH,#84H
    - MOV = 075H
    - DPH = 083H 
      ```
      Executed 0x0000: MOV 83H,#84H | T
      ```
    - 075H 083H #84H
  - b. JNB ACC.0,$
    - JNB = 030H
    - ACC.0 = 0E0H 
      ```
      Executed 0x0000: JNB 0E0H,0FDH |
      ```
    - $ = 0FDH
    - 030H 0E0H 0FDH
  - c. POP DPH
    - POP = 0d0H
    - DPH = 083H 
      ```
      Executed 0x0000: POP 83H |
      ```
    - 0D0H 083H
  - d. MOV A,#'='
    - MOV A = 074H
    - #'=' = #3DH 
      ```
      Executed 0x0000: MOV A,#3DH |
      ```
    - 074H #3DH
- 3.4 What instructions are represented by the following machine language bytes?
  - a. 0EFH
    - MOV A, R7 

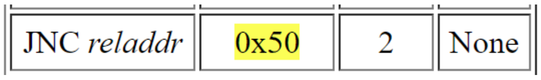
MOV A,R7	0xEF	1	None

  - b. 012H, 080H, 050H
    - LCALL 

LCALL *code addr*	0x12	3	None

    - SJMP 

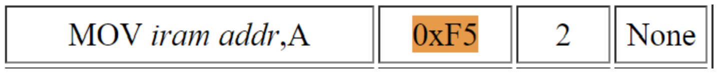
SJMP *reladdr*	0x80	2	None

    - JNC 

JNC *reladdr*	0x50	2	None

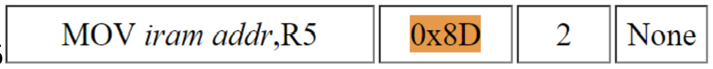
    - LCALL SJMP JNC
  - c. 0F5H, 08DH

        - MOV __, A



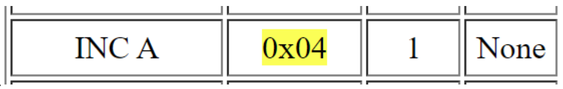
MOV *iram addr*,A	0xF5	2	None

        - MOV __, R5



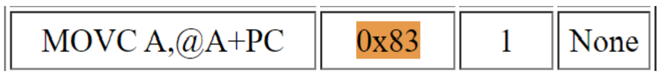
MOV *iram addr*,R5	0x8D	2	None

    - d. 004H
        - INC A



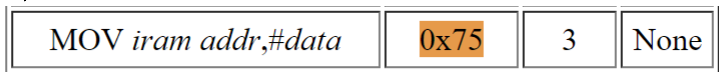
INC A	0x04	1	None

    - e. 083H
        - MOVC A, @A+PC



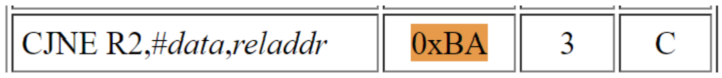
MOVC A,@A+PC	0x83	1	None

    - f. 075H, 0BAH, 0E7H
        - MOV



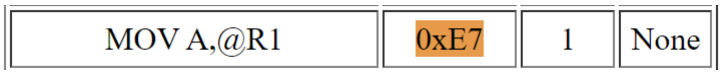
MOV *iram addr*,*#data*	0x75	3	None

        - CJNE



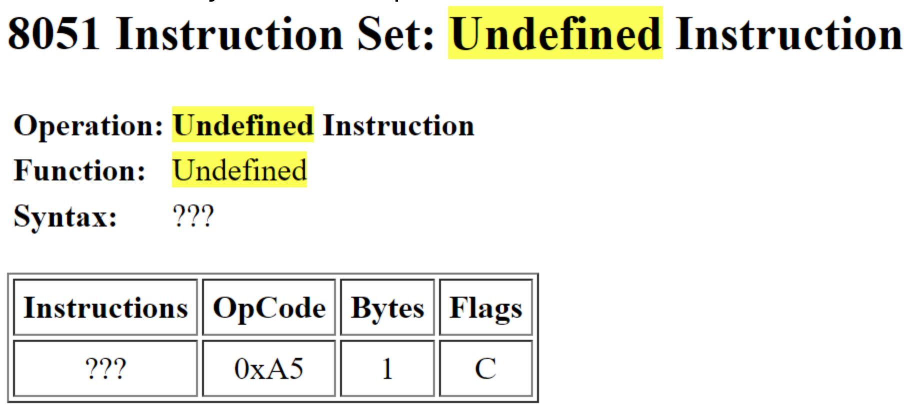
CJNE R2,*#data*,*reladdr*	0xBA	3	C

        - MOV A, @R1



MOV A,@R1	0xE7	1	None

- 3.9 What opcode is undefined on the 8051?
    - 0A5H is the only undefined opcode on the 8051



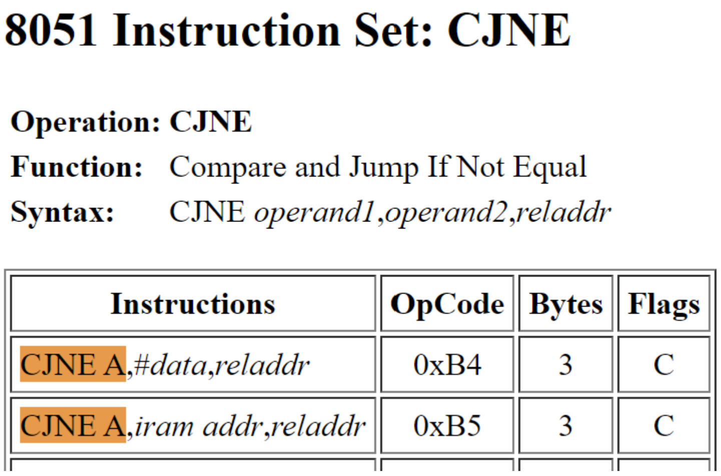
8051 Instruction Set: Undefined Instruction

Operation: Undefined Instruction
Function: Undefined
Syntax: ???

Instructions	OpCode	Bytes	Flags
???	0xA5	1	C

- 3.12 The following is an 8051 instruction: CJNE A,#’Q’, AHEAD
    - What is the opcode for this instruction?

        - The opcode for this instruction is 0xB4, because it does not take anything from an IRAM address

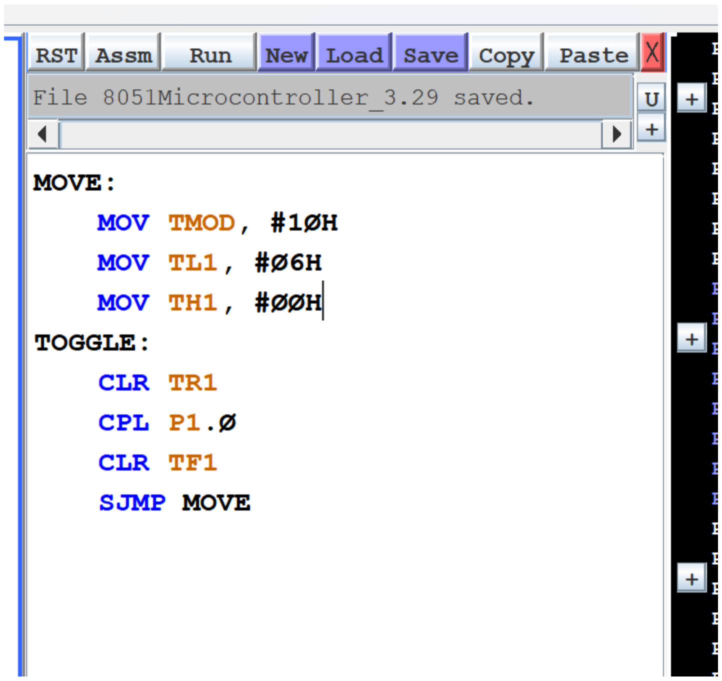


8051 Instruction Set: CJNE

Operation: CJNE
Function: Compare and Jump If Not Equal
Syntax: CJNE *operand1,operand2,reladdr*

Instructions	OpCode	Bytes	Flags
CJNE A,*#data,reladdr*	0xB4	3	C
CJNE A,*iram addr,reladdr*	0xB5	3	C

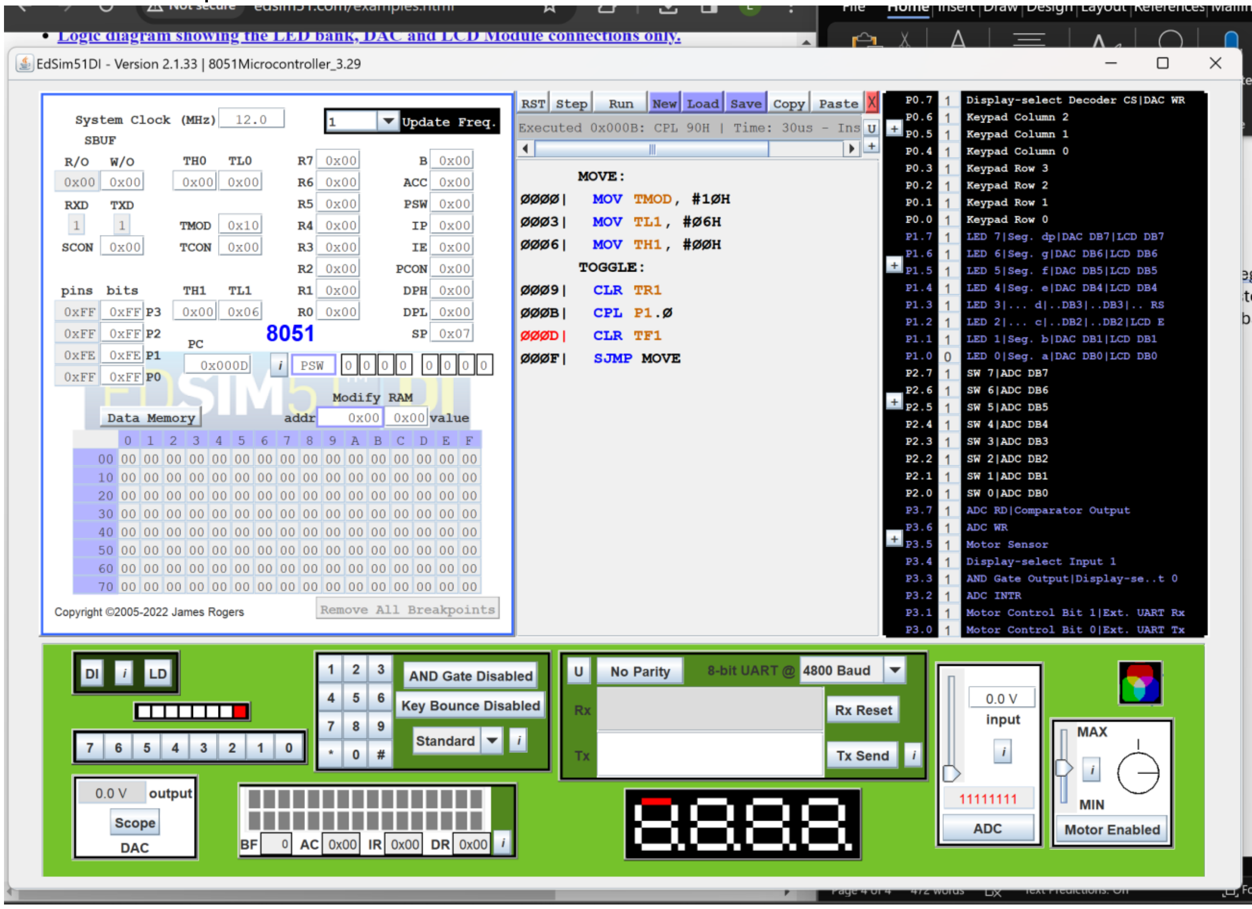
    - How many bytes long is this instruction?
        - This instruction is 3 bytes long
    - Explain the purpose of each byte of this instruction
        - The first byte of the instruction indicates that the function being called is the “compare & jump if !=” function, specifically with one of the operands being the accumulator (register A)
        - The second byte is the second of data being compared (operand2). It can either be raw data or an instruction RAM address, but in this case it is raw data
        - The third byte is indicating the relative address if the two operands are not equal to one another
    - How many machine cycles are required to execute this instruction?
        - This instruction needs two cycles to be executed
            - Page 17 of PDF: https://www.keil.com/dd/docs/datashts/atmel/at_c51ism.pdf
    - If an 8051 is operating from a 10mhz crystal, how long does this instruction take to execute?
        - 8051 needs 12 clock cycles/machine cycle
        - Page 6 of PDF agrees with 12 clock cycles*2 totaling 24: https://www.keil.com/dd/docs/datashts/atmel/at_c51ism.pdf
        - 10mhz → 10mil cycle/sec
        - T=24/10,000,000 = 0.0000024sec → 2.4microsec
- 3.29 Write a program to create an 83.3kHz square wave on P1.0. (Assume 12mhz operation.)



File 8051Microcontroller_3.29 saved.

```
MOVE:
    MOV  TMOD, #1ØH
    MOV  TL1, #Ø6H
    MOV  TH1, #ØØH
TOGGLE:
    CLR  TR1
    CPL  P1.Ø
    CLR  TF1
    SJMP MOVE
```

-
- This program toggles P1.0 (also part of the 7 segment display) every 7 cycles
- It first turns on at the 5th click of the 'step' button, then it turns off at the 12th click of the step button. It then turns back on on the 19th click



EdSim51DI - Version 2.1.33 | 8051Microcontroller_3.29

System Clock (MHz) 12.0 | Update Freq. 1

SBUF R/O	SBUF W/O	TH0	TL0	TMOD	TCON	TH1	TL1	SCON	RXD	TXD	PC
0x00	0x00	0x00	0x00	0x10	0x00	0x00	0x06	0x00	1	1	0x000D

	pins	bits
P3	0xFF	0xFF
P2	0xFF	0xFF
P1	0xFE	0xFE
P0	0xFF	0xFF

R7	0x00	B	0x00
R6	0x00	ACC	0x00
R5	0x00	PSW	0x00
R4	0x00	IP	0x00
R3	0x00	IE	0x00
R2	0x00	PCON	0x00
R1	0x00	DPH	0x00
R0	0x00	DPL	0x00
		SP	0x07

PSW 0 0 0 0 0 0 0 0

Modify RAM addr 0x00 value 0x00

Data Memory	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
10	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
20	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
30	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
40	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
50	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
60	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
70	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Copyright ©2005-2022 James Rogers

Executed 0x000B: CPL 90H | Time: 30us - Ins

```
      MOVE:
ØØØØ|     MOV  TMOD, #1ØH
ØØØ3|     MOV  TL1, #Ø6H
ØØØ6|     MOV  TH1, #ØØH
      TOGGLE:
ØØØ9|     CLR  TR1
ØØØB|     CPL  P1.Ø
ØØØD|     CLR  TF1
ØØØF|     SJMP MOVE
```

P0.7	1	Display-select Decoder CS|DAC WR
P0.6	1	Keypad Column 2
P0.5	1	Keypad Column 1
P0.4	1	Keypad Column 0
P0.3	1	Keypad Row 3
P0.2	1	Keypad Row 2
P0.1	1	Keypad Row 1
P0.0	1	Keypad Row 0
P1.7	1	LED 7|Seg. dp|DAC DB7|LCD DB7
P1.6	1	LED 6|Seg. g|DAC DB6|LCD DB6
P1.5	1	LED 5|Seg. f|DAC DB5|LCD DB5
P1.4	1	LED 4|Seg. e|DAC DB4|LCD DB4
P1.3	1	LED 3|... d|..DB3|..DB3|.. RS
P1.2	1	LED 2|... c|..DB2|..DB2|LCD E
P1.1	1	LED 1|Seg. b|DAC DB1|LCD DB1
P1.0	0	LED 0|Seg. a|DAC DB0|LCD DB0
P2.7	1	SW 7|ADC DB7
P2.6	1	SW 6|ADC DB6
P2.5	1	SW 5|ADC DB5
P2.4	1	SW 4|ADC DB4
P2.3	1	SW 3|ADC DB3
P2.2	1	SW 2|ADC DB2
P2.1	1	SW 1|ADC DB1
P2.0	1	SW 0|ADC DB0
P3.7	1	ADC RD|Comparator Output
P3.6	1	ADC WR
P3.5	1	Motor Sensor
P3.4	1	Display-select Input 1
P3.3	1	AND Gate Output|Display-se..t 0
P3.2	1	ADC INTR
P3.1	1	Motor Control Bit 1|Ext. UART Rx
P3.0	1	Motor Control Bit 0|Ext. UART Tx

AND Gate Disabled | Key Bounce Disabled | Standard | No Parity | 8-bit UART @ 4800 Baud | DAC output 0.0 V | ADC input 0.0 V | 11111111 | Motor Enabled | BF 0 | AC 0x00 | IR 0x00 | DR 0x00

-