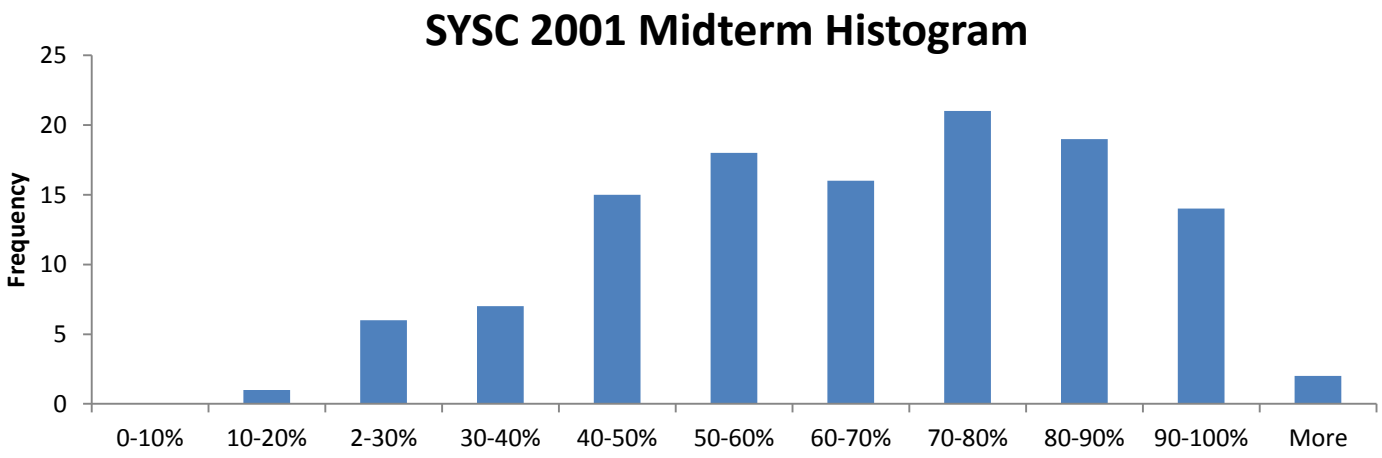
SYSC2001 Midterm Solutions

25 Oct 2012

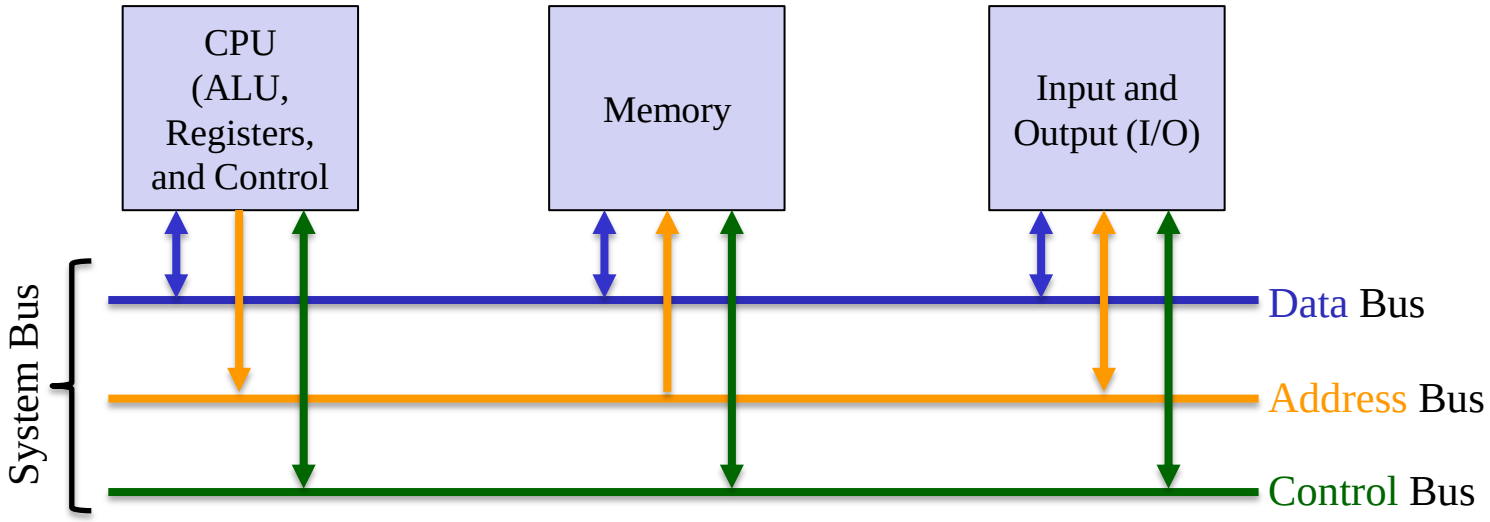
Class Performance

- Average = 66% (23.81/36)
- Median = 67% (24/36)



1a) The System Bus Model

- The system bus model is a refinement of the von Neumann model:
  - CPU (ALU and Control), memory, and input/output unit.
- The *system* bus provides communication among components
  - Made up of: data bus, the address bus, and the control bus.
  - Optional: *power bus*, separate *I/O bus*.



Adapted from [2]

1b) Fetch-Execute Cycle

- IP points to next instruction to be fetched
- Fetched from memory into IR, which holds instruction currently being executed
- Executed from IR
- Increment IP
- Repeat

1c) p86 Assembly Language: 2 Pass Assembly Process

- **each pass: processes all statements in .ASM file sequentially from start to finish**

1st Pass: for each statement:

1. **check syntax**
2. **allocate any memory needed for image**
   - **memory declaration (DB, DW)**
   - **instruction: opcode + operands**
3. **if encounter a label definition: assign value to label and keep record of (label, value) association in Symbol Table**

- **if syntax errors in 1st pass, then write errors to .LST file and stop, else ....**

2nd Pass: build binary for each statement:

- **may require calculating offsets – may result in errors – e.g. trying to jump too far for a conditional jump (target out of range)**
- **write results to .LST file**
- **if no errors – write results to .OBJ file**

2

```
DISP EQU 04E9h

STR DB 'Bonjour Monde!@'

; Function printStr
; Prints a '@'-terminated string
; Input parameters:
;           BX: starting address of string
; Output parameters:
;           None.
printStr:
            PUSH AX
            PUSH BX
            MOV DX, DISP
Loop:
            MOV AL,[BX]                         ; FIX ME (read the next char from memory)
            CMP AL, '@'                         ; FIX ME (compare char to '@')
            JE quit                             ; FIX ME (jump to quit if you find an '@')
            OUT [DX],AL                         ; FIX ME (print the character)
            INC BX
            jmp Loop
quit:
            POP BX                              ; FIX ME (restore the registers)
            POP AX                              ; FIX ME (restore the registers)
            RET

main:
            MOV BX,STR                          ; FIX ME (set input parameter for printStr)
            CALL printStr
            HLT
END main
```

Q3

Initial:

Address	Value
1000	A5h
1001	43h
1002	9Bh
1003	08h
1004	0Ch

AX: 1234h	SI: 0005h
BX: 7B9Ah	DI: 99FFh
CX: ABCDh	BP: 0000h
DX: 1003h	SP: FFFEh

MOV CL, [DX]:

Address	Value
1000	
1001	
1002	
1003	
1004	

AX:	SI:
BX:	DI:
CX: AB08h	BP:
DX:	SP:

MOV BX, 1001h :

Address	Value
1000	
1001	
1002	
1003	
1004	

AX:	SI:
BX: 1001h	DI:
CX:	BP:
DX:	SP:

MOV AH, [1001h] :

Address	Value
1000	
1001	
1002	
1003	
1004	

AX: 4334h	SI:
BX:	DI:
CX:	BP:
DX:	SP:

MOV [BX+1], CX:

Address	Value
1000	
1001	
1002	08h
1003	ABh
1004	

AX:	SI:
BX:	DI:
CX:	BP:
DX:	SP:

Q4

a) $1Ah = 1*16^1 + 10*16^0 = 26_{10}$ or $2^4+2^3+2^1 = 26$

b) -47_{10}: first find binary of $+47$, then change sign

- Use remainder method to find $+47_{10} = 00010111_2$
- Apply 2's comp procedure to change sign: 11010001_2

c) 4Ch + 75h = C1h = 1100 0001

C=0, V=1, S=1, Z=0

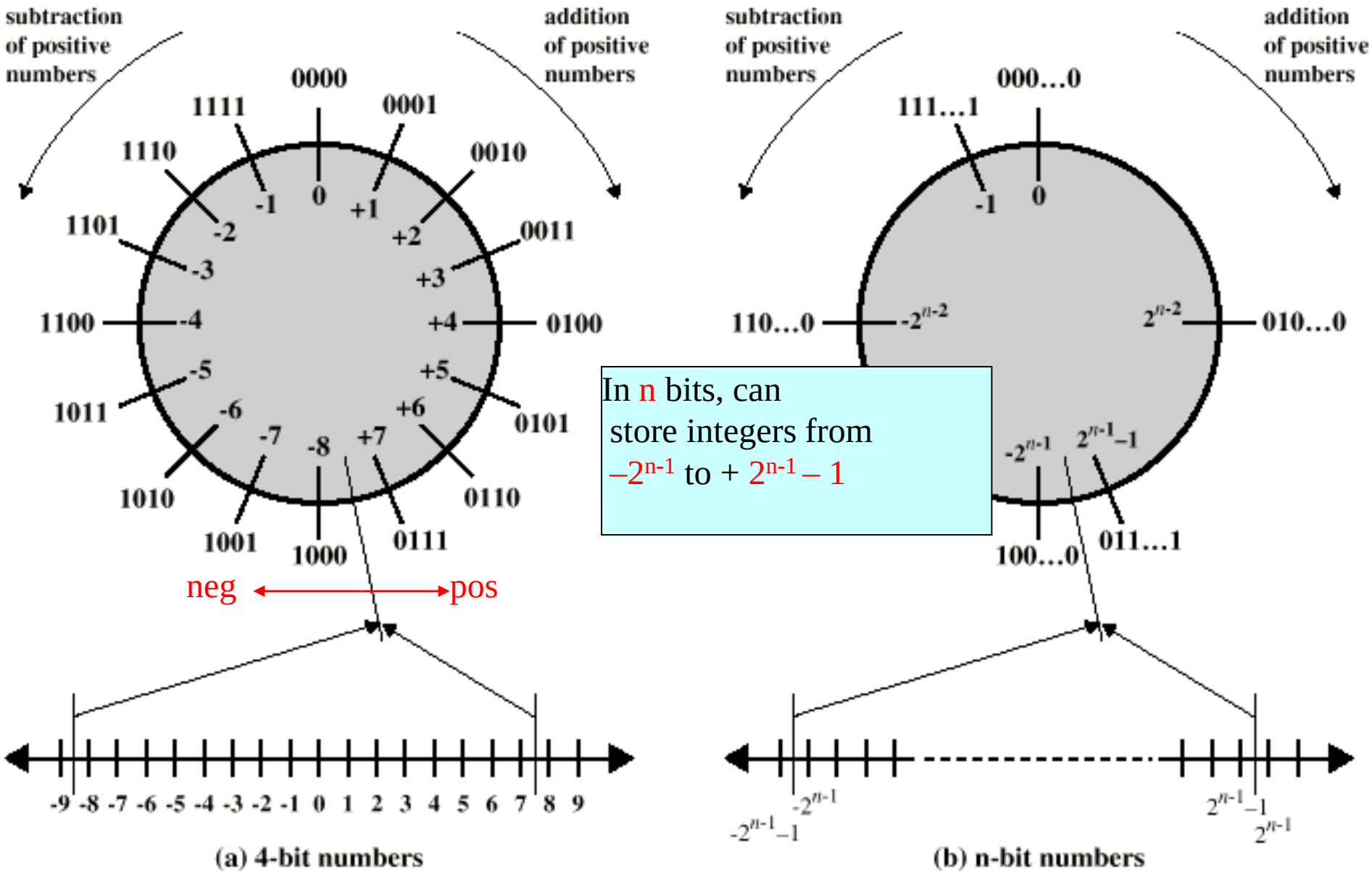
d) 6-9=-3=11111101_2 Compute as +6 + (-9) = 00000110

+11110111

C=0, V=0, S=1, Z=0 =11111101

So what is "2's complement representation" anyways?

Two's Complement Integers



In n bits, can store integers from -2^{n-1} to $+ 2^{n-1} - 1$

Range of Numbers

- 8 bit 2s complement
  - +127 = 01111111 = 2^7 -1
  - -128 = 10000000 = -2^7
- 16 bit 2s complement
  - +32767 = 01111111 11111111 = 2^{15} - 1
  - -32768 = 10000000 00000000 = -2^{15}
- N bit 2s complement
  - 011111111..11111111 = 2^{N-1} - 1
  - 100000000..00000000 = -2^{N-1}

Largest positive

Smallest (?) negative

Example: 8-bit signed number representation

Decimal	Sign&Mag	1's Comp.	2's Comp
0	00000000 or 10000000	00000000 or 11111111	00000000
127	01111111	01111111	01111111
-127	11111111	10000000	10000001
56	00111000	11000111	11001000
-25	10011001	11100110	11100111
128	NA	NA	NA
-128	NA	NA	10000000
200	NA	NA	NA

Q5a)

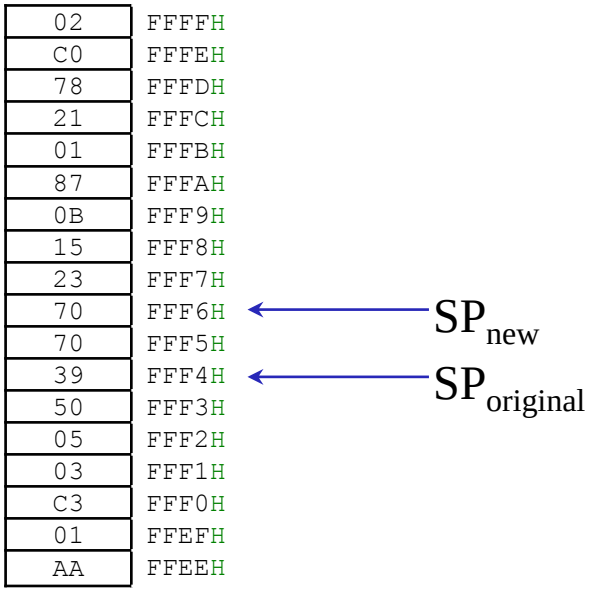
- `CALL target` does 2 things:
  - 1) Push `IP` onto the stack to save the return address
  - 2) Change the `IP` to point to `target` (or add `target` to `IP`)

Q5b)

Registers before `POP SI`:

AX	00A5h
BX	A400h
CX	1000h
DX	0560h

SI	43A1h
DI	0050h
SP	FFF4h
IP	2203h



Registers after `POP SI`:

AX	
BX	
CX	
DX	

SI	7039h
DI	
SP	FFF6h
IP	2204h

Remarking Requests

- Put your request in writing.
- Resubmit request with midterm
- Entire midterm will be reexamined
- A change of 0.5 marks leads to your final grade changing by 0.7%