# ELEG 3230B Tutorial 4

**Yeung Chue Hei**

**SHB 827**

# Outline

- Instruction set
- Program Example
- 8088 Pin Assignment
- 8284 Clock Generator

# Difference between CMP and TEST

- CMP is a subtraction that changes only the flag bits (Usually for comparing numbers)
- SUB CL, BL ; CL = CL – BL
- CMP CL, BL ; CL – BL

- TEST performs AND operation that changes only the flag bits (Usually for checking individual bits)
- AND AL, BL ; AL = AL and BL
- TEST AL, BL ; AL and BL

- Usually followed by conditional jump

# Control Transfer Instructions

- Conditional JUMP

| | Assembly Language | Tested Condition | Operation |
|---|---|---|---|
| For unsigned number | JA | Z = 0 and C = 0 | Jump if above |
| | JAE | C = 0 | Jump if above or equal |
| | JB | C = 1 | Jump if below |
| | JBE | Z = 1 or C = 1 | Jump if below or equal |
| | JC | C = 1 | Jump if carry |
| | JE or JZ | Z = 1 | Jump if equal or jump if zero |
| For signed number | JG | Z = 0 and S = 0 | Jump if greater than |
| | JGE | S = 0 | Jump if greater than or equal |
| | JL | S != O | Jump if less than |
| | JLE | Z = 1 or S != O | Jump if less than or equal |
| | JNC | C = 0 | Jump if no carry |
| | JNE or JNZ | Z = 0 | Jump if not equal or jump if not zero |
| | JNO | O = 0 | Jump if no overflow |
| Use for both | JNS | S = 0 | Jump if no sign (positive) |
| | JNP or JPO | P = 0 | Jump if no parity or jump if parity odd |
| | JO | O = 1 | Jump if overflow |
| | JP or JPE | P = 1 | Jump if parity or jump if parity even |
| | JS | S = 1 | Jump if sign (negative) |
| | JCXZ | CX = 0 | Jump if CX is zero |

# Example

- A short program that stores the content of AL only if it is greater than 04h

- ```
  CMP       AL, 04h
  JLE       CONTINUE ; jump if AL is less than
                     ; or equal to 04h
  MOV       BL, AL
  CONTINUE:
  …
  …
  ```


  ```

# Example

- A short program that tests the rightmost and leftmost bit positions of the AL register.

```
TEST    AL, 1       ; 1d = 00000001b
JNZ     RIGHT       ; jump if rightmost bit is
                    ; not zero

TEST    AL, 128     ; 128d = 10000000b
JNZ     LEFT        ; jump if leftmost bit is
                    ; not zero
```

# LOOP

- LOOP
  - = DEC CX
    JNZ

- (If CX != 0)
- LOOPE Loop while equal
- LOOPZ Loop while Z = 1

- LOOPNE Loop while not equal
- LOOPNZ Loop while Z = 0

# Example

- Calculate 1+2+3+4+5+6+7+8+9+10

- ```
  sum = 0;
  for(i = 10 ; i > 0 ; i--)
  sum += i
  ```

- ```
  MOV       AX, 0
  MOV       CX, 10
  L1:
        ADD   AX, CX
        LOOP L1
  ```

# Example

- ```
  MOV     CX, 0Ah
  MOV     AX, 7h
  L1:
  AND     AX, CX
  LOOPNZ  L1
  ```
- After operation, AX=? CX=?


  ```

# Processor Control Instructions

- Carry Flag Control
- STC Set carry
- CLC Clear carry
- CMC Complement carry

- Interrupt Control
- STI Set interrupt flag
- CLI Clear interrupt flag

- Direction Flag Control
- STD Set direction flag
- CLD Clear direction flag

# Processor Control Instructions

- NOP
  - No operation
  - Use to insert time delays to waste time.

# Exercise

- Assume a device measures the temperature T and store the value in AL.
- Its lower 7-bit value (0 to 127) corresponds to a temperature reading from 0 to 127°C.
- Write a segment of program to store a value 10, 20 and 30 in BL, respectively, if the temperature reading T is in the following three ranges: T < 40°C , 40°C ≦ T<80°C, 80°C ≦ T.

# Answer

```
1.  CMP     AL, 28H     ; 28H = 40d
2.  JL      BOUT10
3.  CMP     AL, 50H     ; 50H = 80d
4.  JL      BOUT20
5.  MOV     BL, 1EH     ; 1EH = 30d
6.  JMP     END
BOUT10:
7.  MOV     BL, 0AH     ; 0AH = 10d
8.  JMP     END
BOUT20:
9.  MOV     BL, 14H     ; 14H = 20d
10. JMP     END
11. END:    NOP
```

# Exercise

- There are total 130 students registered for the course ELE3230B. Assume their last 4-digit students' ID are stored at Data Segment starting at offset address 2000h and beyond, with less significant digit in lower order byte. For example, if the first student's ID is 06232781, then 2781h is stored in [2000h] and [2001h]. You need to write a segment of program to find all students' ID with the last digit of their students' ID equal to 4 and put those students' ID in Extra segment, at offset address 3000h and beyond.

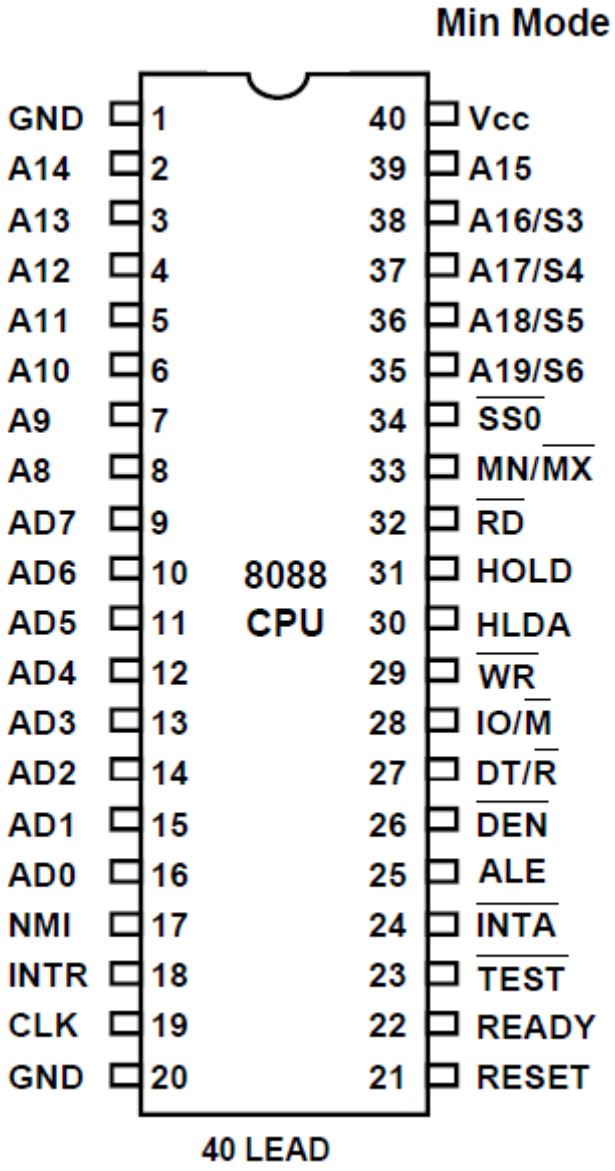
- Write the segment of program.

# Answer

```
MOV  SI, 2000H
MOV  DI, 3000H
MOV  CX, 130

LABLE1:
MOV  AL, DS:[SI]
AND  AL, 0FH
CMP  AL, 04H
JNE  CONTINUE
```

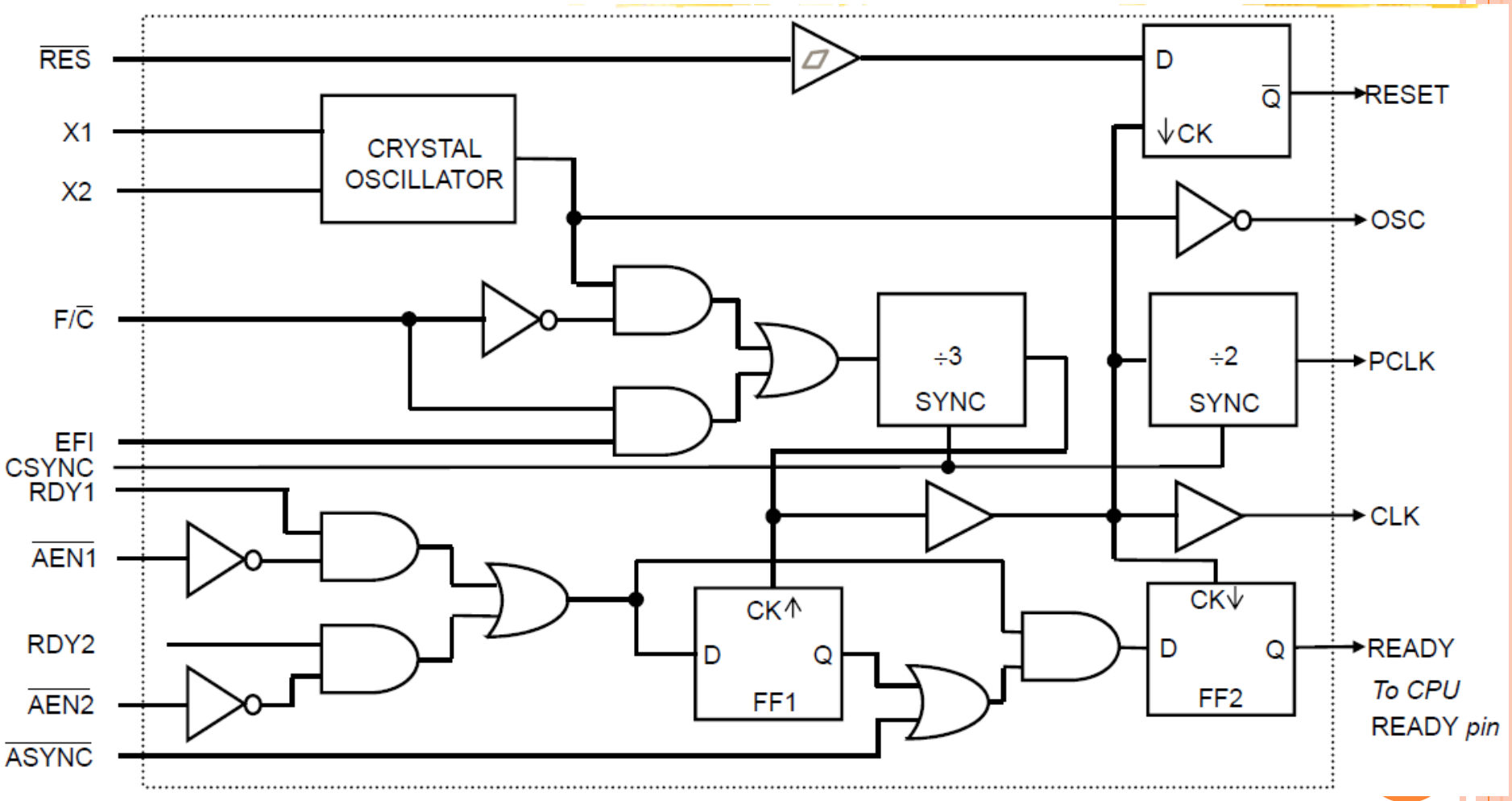
```
MOV  BX, DS:[SI]
MOV  ES:[DI], BX
ADD  DI, 02H

CONTINUE:
ADD  SI, 02H

LOOP LABLE1
```

# 8088 Pin Assignment




1. Power Supply and Clock (VCC, GND and CLK)
2. Minimum/Maximum Mode pin (MN/$\overline{MX}$)
3. Bus Master pins (HOLD, HLDA)
4. Interrupt pins (NMI, INTR and $\overline{INTA}$)
5. RESET
6. Bus control pins ($\overline{RD}$, ALE, $\overline{DEN}$, DT/$\overline{R}$, $\overline{WR}$, IO/$\overline{M}$)
7. Address, data pins and address status pins

# 8284 Clock Generator



# Output

- OSC (Oscillator)
  - At the crystal frequency
  - To the EFI pin of another 8284
- CLK (Clock)
  - At 1/3 the EFI frequency or crystal frequency
  - To the CLK pin of 8088
- PCLK (Peripheral clock)
  - At 1/2 the CLK frequency
  - To peripheral ICs
- READY
  - To the READY pin of 8088
- RESET
  - To the RESET pin of 8088

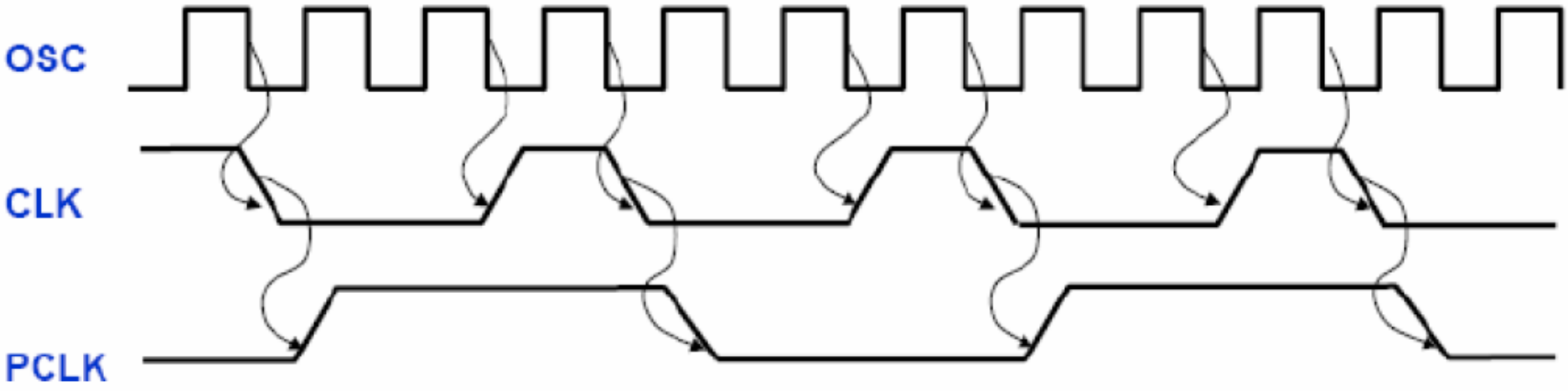
# Input

- Power supply
  - VCC, GND
- To generate OSC, CLK, PCLK
  - X1, X2 (Crystal inputs)
    - Connected with an external crystal
  - EFI (External Frequency Input)
    - From the OSC pin of another 8284
  - CSYNC (Clock Synchronization)
    - Used with EFI; Grounded if crystal is used
  - F/C (Frequency/Crystal)
    - Select EFI or crystal

# Input

- To generate READY
  - RDY1, RDY2 (bus ready)
  - AEN1, AEN2 (address enable)
    - Qualify RDY1, RDY2
  - ASYNC (ready synchronization)
    - Select 1 or 2 stages of synchronization
- To generate RESET
  - RES

- Relationship among OSC, CLK and PCLK




| clock | Generate by | Used for | Example value |
|---|---|---|---|
| oscillator | crystal | 8284 | 15MHz |
| CLK | 8284 | 8088 | 5MHz |
| PCLK | 8284 | Peripheral ICs | 2.5MHz |

# Summary

- Instruction set
  - Arithmetic and Logic Instructions
    - CMP and TEST
  - Control Transfer Instructions
    - Conditional Jump
  - Processor Control Instructions
    - STC, CLC, CMC
    - STI, CLI
    - NOP
- 8088 Pin Assignment
- 8284 Clock Generator

# END