CE320 : Microcomputers I

Lab4:
Code conversion

Jung Me Park, Ph.D

ECE Department, Kettering University

jpark@kettering.edu

Kettering UNIVERSITY | College of Engineering

Binary Coded Decimal (BCD)

- Although all computers work internally with binary numbers,
  - the input and output equipment generally uses decimal numbers.
  - the decimal numbers must be coded in terms of binary signals.

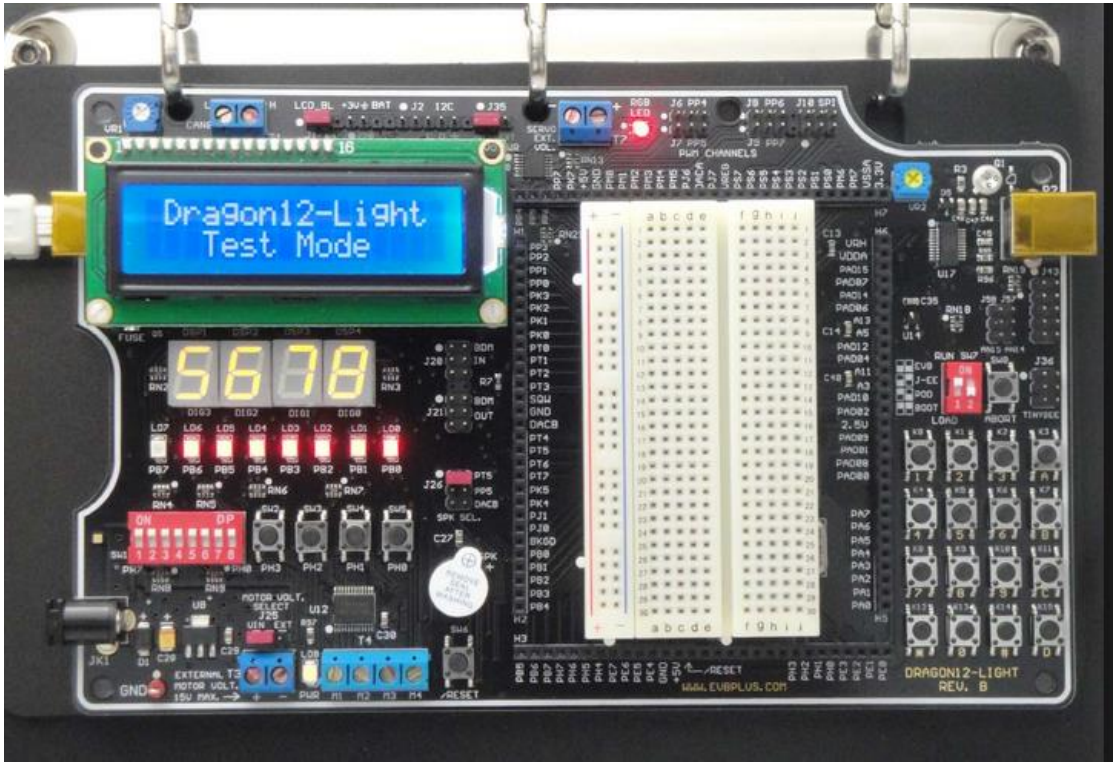
- A binary coded decimal (BCD):
  - In the simplest form of binary code, each decimal digit is represented by its binary equivalent.
    - For example, **5,678** is represented by

5	**6**	**7**	**8**
0101	**0110**	**0111**	**1000**

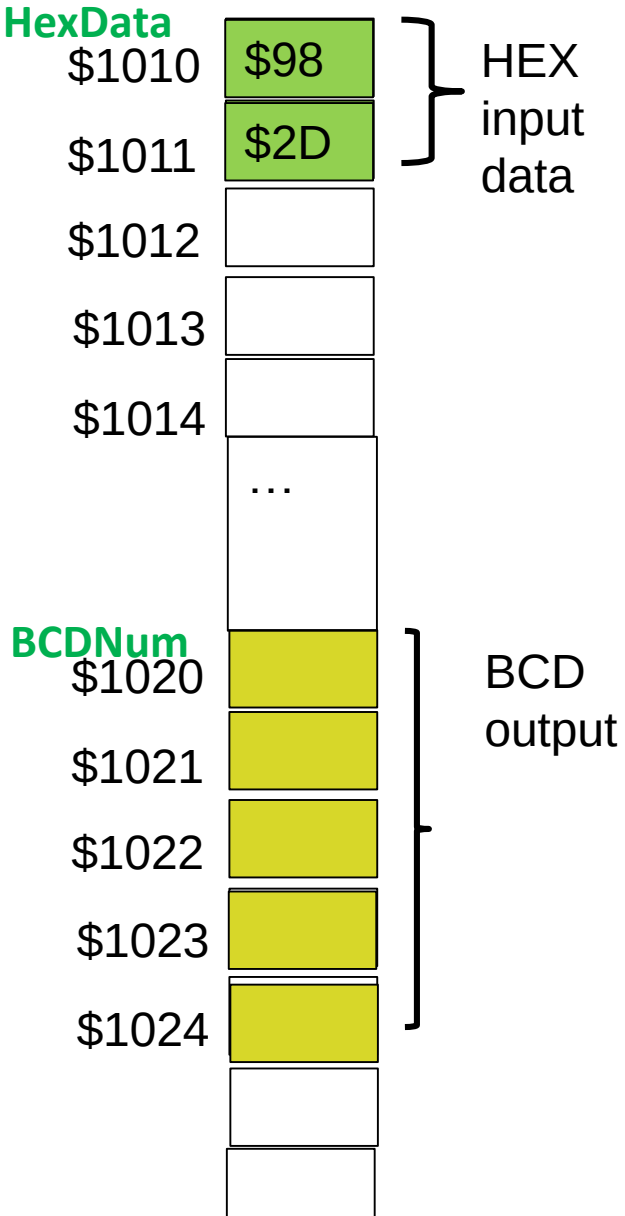
  - This representation is called **a Binary Coded Decimal (BCD)**

Binary Coded Decimal (BCD)

- Binary Coded Decimal (BCD) is a very convenient way of representing numbers that will be sent to external 7-segment displays.



Lab4: Data Allocation in Lab4

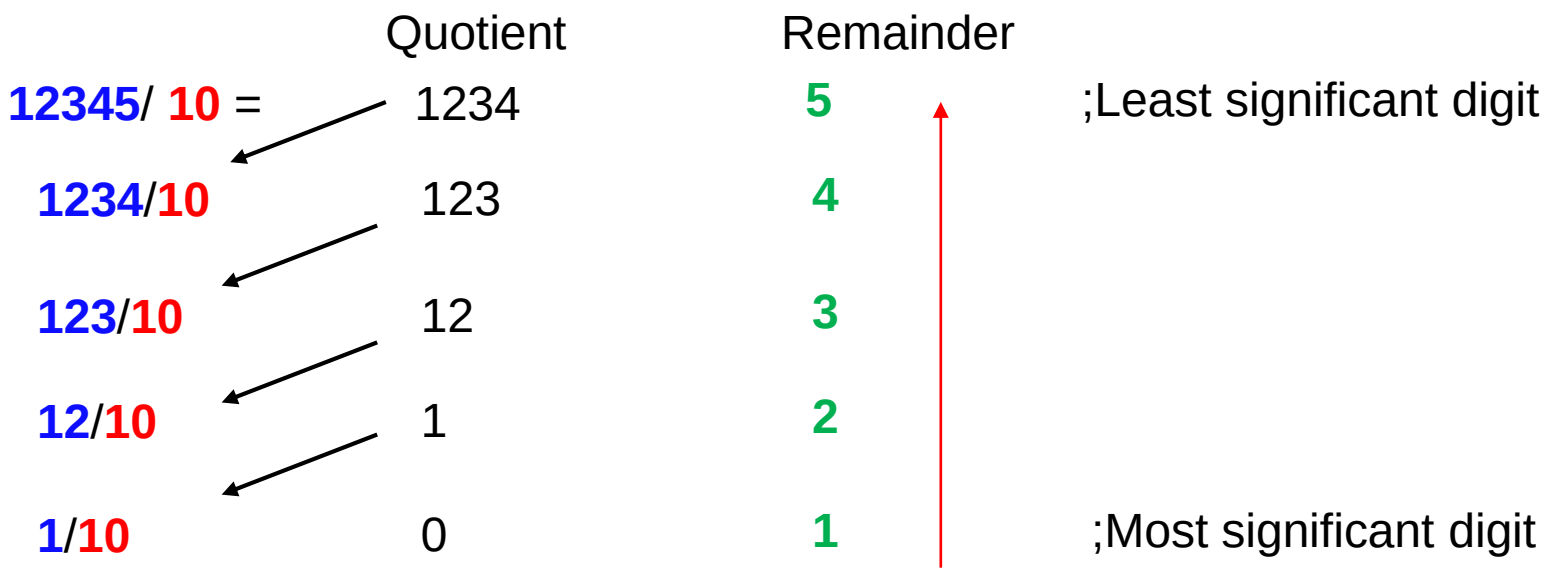



```
                ORG $1010
HexData:    DC.W    $982D   ; 2-byte  input data
```

```
                ORG $1020
BCDNum:     DS.B    5       ; reserve 5-byte for the BCD output
```

Binary to BCD Conversion

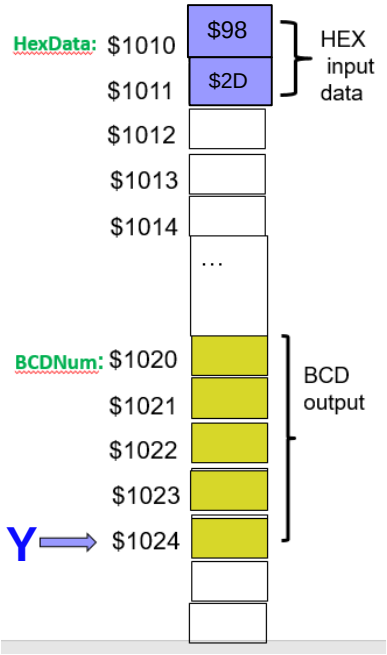
- A binary number can be converted to BCD format **using repeated division by 10.**
  - The number is divided by 10, the remainder is recorded
  - The quotient is then divided by 10, the remainder is recorded
  - **The process is repeated until the quotient is zero**
- Ex:



Division Instructions

Assembly Instruction	Meaning	Operation :Important!
IDIV	**Integer DIV**ision 16 bit by 16 bit unsigned Integer Division	[D] ÷ [X] ⇒ X=Quotient D=Remainder
IDIVS	**Integer DIV**ision 16 bit by 16 bit **Signed** integer Division	[D] ÷ [X] ⇒ X=Quotient D=Remainder
EDIV	Extended **DVI**sion unsigned 32 bit by 16 bit division	[Y :D] ÷ [X] ⇒ Y=Quotient D=Remainder
EDIVS	**E**xtended **DVI**sion Signed 32 bit by 16 bit division	[Y :D] ÷ [X] ⇒ Y=Quotient D=Remainder

Binary to BCD Conversion



- Pseudocode: How to convert Binary to BCD

1. Load the address of BCD for **the least significant digit**.

   LDY # BCDNum +4

2. Load the 16 bit number into register D
3. Compare register Y with **# BCDNum**

   CPY # BCDNum

4. If register **Y < # BCDNum**, then exit the loop

   BLO ExitLoop ; ExitLoop is a label

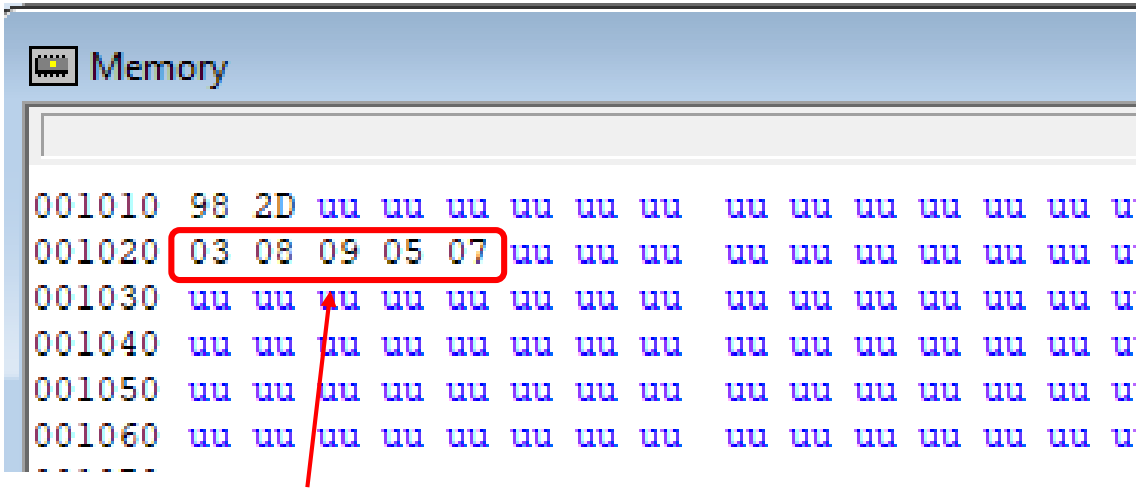
5. Load #10 into register **X** **; register X is a divider**
6. **IDIV** ; register **D / register X**

   i) Quotient is saved in register X, ii) Remainder is saved in register D (B register inside D)

7. Save the remainder (register B) at the memory location in **Y**
8. **XGDX** :Exchange D with X ; → so D can hold the Quotient for the next division
9. Decrement register Y by 1
10. BRA Step3.

BCD Output

- The BCD output will be displayed at the memory location, $1020:1024. Take the screen shots of the BCD output in the memory pane, Assembly pane and Register pane.

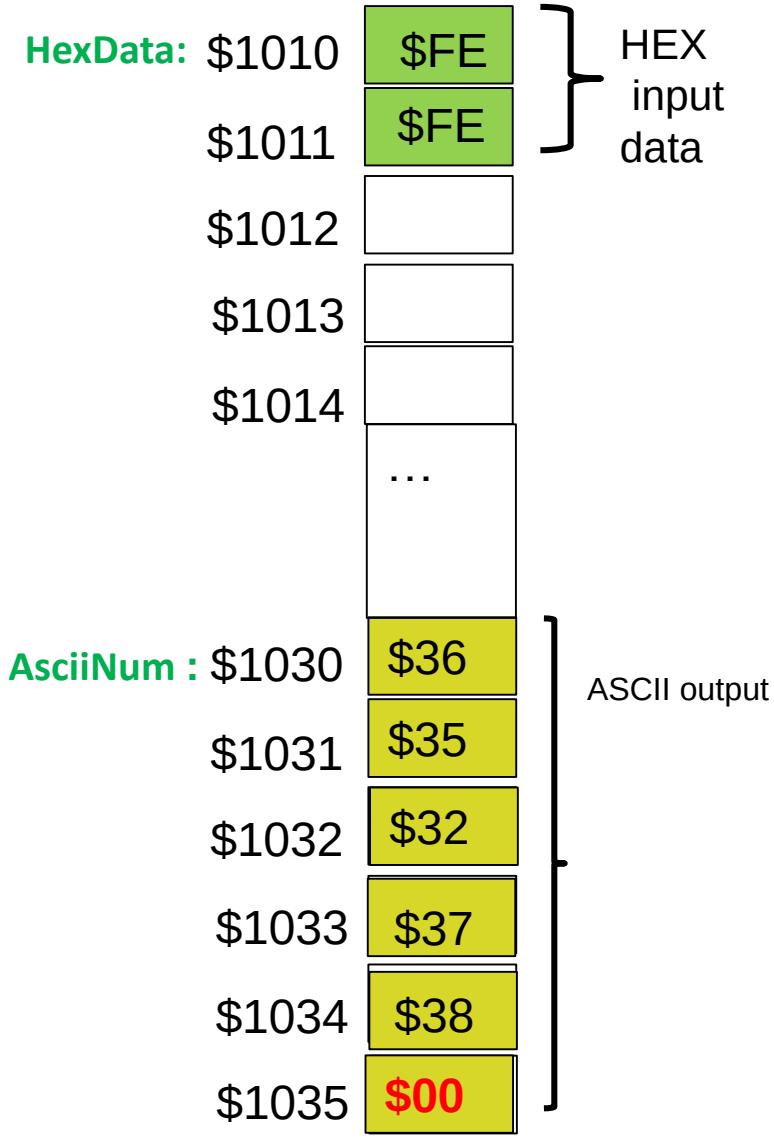


BCD result at $1020:1024

ASCII CODE CONVERSION

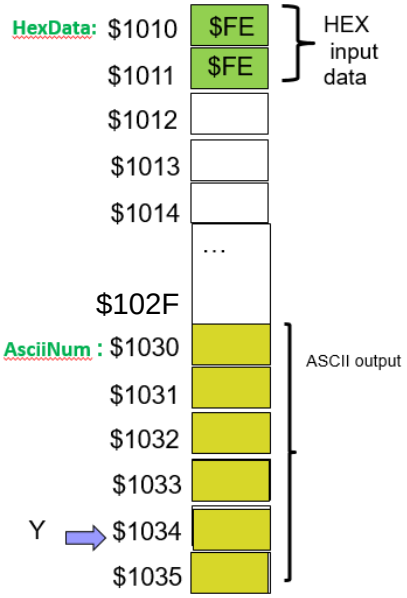
jpark@kettering.edu Kettering UNIVERSITY | College of Engineering

Binary to BCD Conversion: Example Code(I)



ASCII code =
$BCD digit + $30

Binary to Ascii Conversion



- Pseudocode: How to convert Binary to BCD

1. Load the address of Ascii for **the least significant digit**.

   LDY # AsciiNum +4

2. Load the 16 bit number into register D
3. Compare register Y with **# AsciiNum**

   CPY # AsciiNum

4. If register **Y < #AsciiNum**, then exit the loop

   BLO ExitLoop ; ExitLoop is a label where the brach is taken

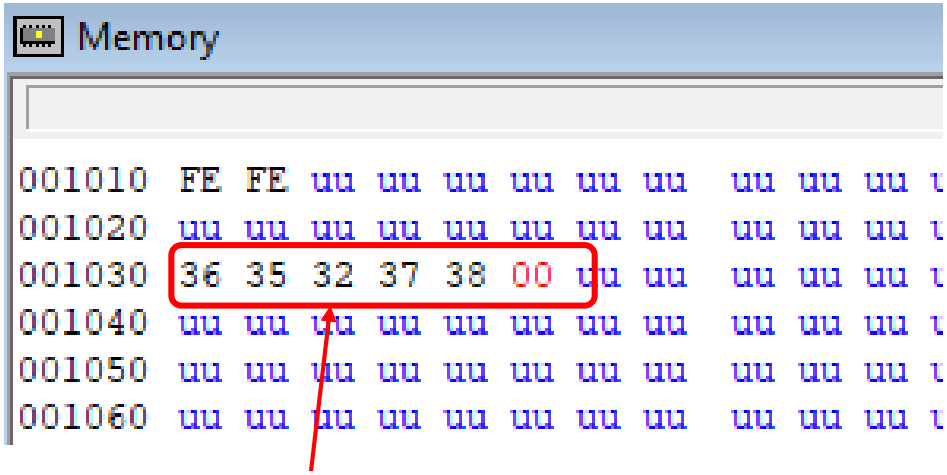
5. Load #10 into register **X** ; **register X is a divider**
6. **IDIV** ; register **D / register X**

   i) Quotient is saved in register X, ii) Remainder is saved in register D (B register inside D)

7. **Add $30 to register B**
8. Save register B at the memory location in register **Y**
9. **XGDX** ;Exchange register D with register X, so D can hold the Quotient
   ; for the next division
7. Decrement register Y register by 1
11. BRA Step3.
12. Move a byte $00 at the memory location **AsciiNum +5 (=$1035)**

ASCII Code Output

- The BCD output will be displayed at the memory location, $1020:1024. Take the screen shots of the BCD output in the memory pane, Assembly pane and Register pane.



ASCII code result at $1030:1034 and ending with a special character, **$00** at $1035