

## TCU CoSc30203§50 - Comp. Sys Fundamentals

=====

Dec	Bin	Hex	Oct
--	----	-	--
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	8	10
9	1001	9	11
10	1010	A	12
11	1011	B	13
12	1100	C	14
13	1101	D	15
14	1110	E	16
15	1111	F	17
16	10000	10	20
17	10001	11	21
18	10010	12	22
19	10011	13	23

20

10100

14

24

## Signed Binary Number Systems

Hex Val	Bin Val	Unsign DecVal	Signed MagDec	One & Two's Complement
-	-----	--	--	--
0	0000	00	+0	+0
1	0001	01	+1	+1
2	0010	02	+2	+2
3	0011	03	+3	+3
4	0100	04	+4	+4
5	0101	05	+5	+5
6	0110	06	+6	+6
7	0111	07	+7	+7
8	1000	08	-0	-7
9	1001	09	-1	-6
A	1010	10	-2	-5
B	1011	11	-3	-4
C	1100	12	-4	-3
D	1101	13	-5	-2
E	1110	14	-6	-1
F	1111	15	-7	-0

To take the one's complement of a binary value you must know the size (in bits) of the original number. Express the original value using its total number of bits. Then simply flip all the bits (i.e. zero bits becoming one bits and the original one bits becoming zero bits).

To take the two's complement of a binary value you take the one's complement and then add 1 to the result.

$$\begin{aligned}
 \text{BDh} = 1011\ 1101\text{b} & \Rightarrow -\text{BDh} = -1011\ 1101\text{b} \\
 & = 0100\ 0010\text{b} \\
 & \quad + 1 \\
 & \quad \text{-----} \\
 & = 0100\ 0011\text{b} \\
 & = 43\text{h} = 64\text{d}
 \end{aligned}$$

Note that you take the one & two's complement of both positive & negative values in the same manner.

To subtract 'Y' from 'X', take the two's complement of 'Y' and add the result to 'X'.

Note for 4-bit, 2's complement binary values  
...

	0100			11100	What is com-
3	0011		3	0011	plement of 0
+2	+0010		-2	-0010	& -8?
--	-----		--	-----	
5	0101		1	10001	

C:\TEESSIDE>e68k p1

[E68K PC-2.2h Copyright (c) Univ Teesside 1989,97]

Address space 0 to ^10485759 (10240 kbytes).

Loading binary file "P1.BIN".

Start addr: 000400, Low: 00000400, High: 0000041B

>he

HELP

.D0 .D1 .D2 .D3 .D4 .D5 .D6 .D7  
.A0 .A1 .A2 .A3 .A4 .A5 .A6 .A7  
.PC .SR .US .SS .C .N .V .X .Z

BF - Block fill	INT - Interrupt simulation
BM - Block move	LO - Load binary file
BR - Set/display brkpts	LOG - Log screen output
BS - Block search	MD - Memory dump
DC - Data conversion	MM - Memory modify
DF - Display registers	MS - Memory set
DU - Dump memory to file	NOBR- Remove breakpoints
GD - Go direct	RE - Reset processor
GO - Exec pgm w/ brkpts	TB - Trace buffer function
GT - Go with temp brkpt	TR - Trace program
HE - Help	QUIT- Return to MS-DOS

For more detailed information type: HE command

HE INFO

HE EXCEPTIONS

HE TRAP#15

>he br

Set & display BReakpoints. ...

Format: BR <brkpt\_1> ... <brkpt\_7> -<count>

The BR command allows a user to insert or remove breakpoints in memory. Associated with each breakpoint is a count. This count specifies how many times the breapointed location can be referenced before a break is taken. If no count is specified the default value is zero. Up to seven breakpoints can be inserted with the BR command if a count is specified, eight otherwise. If BR is used with no options the existing breakpoint table is printed. Up to sixteen\breakpoints can be in force at once.

E.G. BR 1000 1020 1030 -2

>he md

Memory Dump. Format: MD <addr> <count> [-DI]

The MD command dumps COUNT bytes of memory ...

>md 400 20

```
000400  43 F9 00 00 04 18 34 11 D4 69 00 02 6E 00 00 04
000410  94 61 4E 72 20 00 00 0F FF FE 00 01 00 00 00 00
000420  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000430  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00.
```

>df

```
PC=000400 SR=2000 SS=00A00000 US=00000000 X=0
A0=00000000 A1=00000000 A2=00000000 A3=00000000 N=0
A4=00000000 A5=00000000 A6=00000000 A7=00A00000 Z=0
D0=00000000 D1=00000000 D2=00000000 D3=00000000 V=0
D4=00000000 D5=00000000 D6=00000000 D7=00000000 C=0
----->LEA.L    $0418,A1
```

>go

Processor halted at: 000416

>df

```
PC=000416 SR=2000 SS=00A00000 US=00000000 X=0
A0=00000000 A1=00000416 A2=00000000 A3=00000000 N=0
A4=00000000 A5=00000000 A6=00000000 A7=00A00000 Z=0
D0=00000000 D1=00000000 D2=0000FFF0 D3=00000000 V=0
D4=00000000 D5=00000000 D6=00000000 D7=00000000 C=0
----->DC.W      $000F
```

## Parameter Passing Mechanics

=====

### Global Variables

- Possible Name Clashes (Conflict)
- Inflexible - Must Always Use Same Vars
- Does Not Lend Itself To Recursion

### Use Registers

- Limited By Number Of Registers
- Not Suitable For Arrays & Structures

### Use Stack

