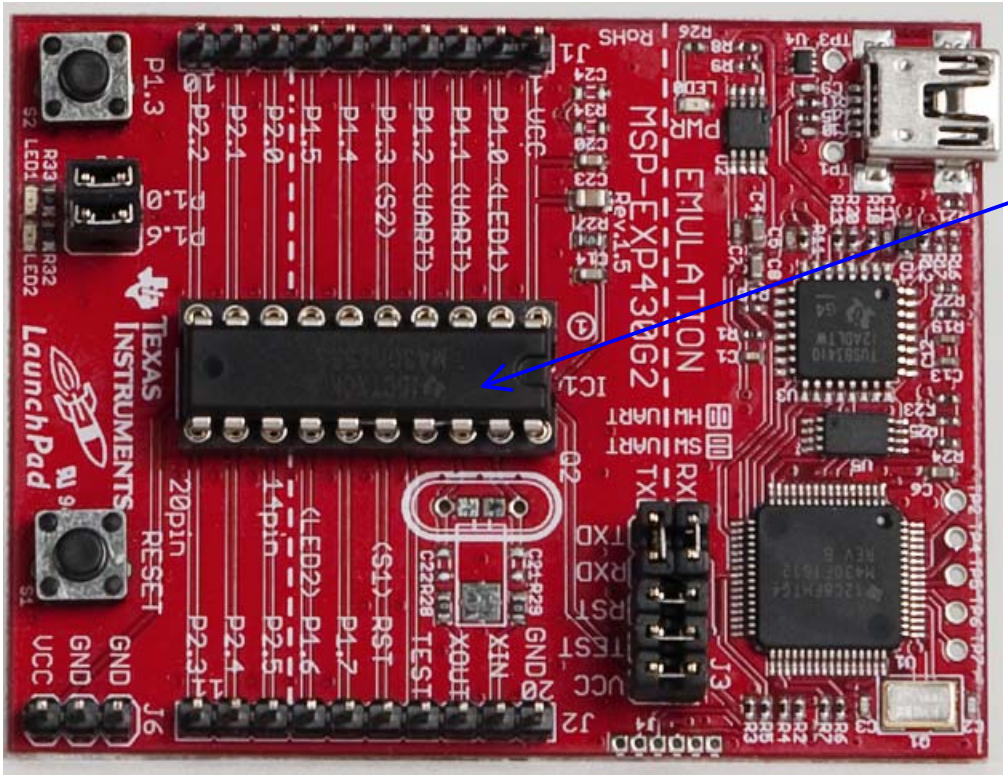
Texas Instruments Launchpad



MSP430 devices have up to eight digital I/O ports implemented, P1 to P8. Number of ports varies with device type and packaging. For MSP430G2553IN20, there are two ports, in total 16 I/O pins available.

Bootstrap Loader

Embedded Emulation Module

Device	BSL	EEM	Flash (KB)	RAM (B)	Timer_A	COMP_A+ Channel	ADC10 Channel	USCI A0/B0	Clock	I/O	Package Type
MSP430G2553IRHB32	1	1	16	512	2x TA3	8	8	1	LF, DCO, VLO	24	32-QFN
MSP430G2553IPW28										24	28-TSSOP
MSP430G2553IPW20										16	20-TSSOP
MSP430G2553IN20										16	20-PDIP

ROM

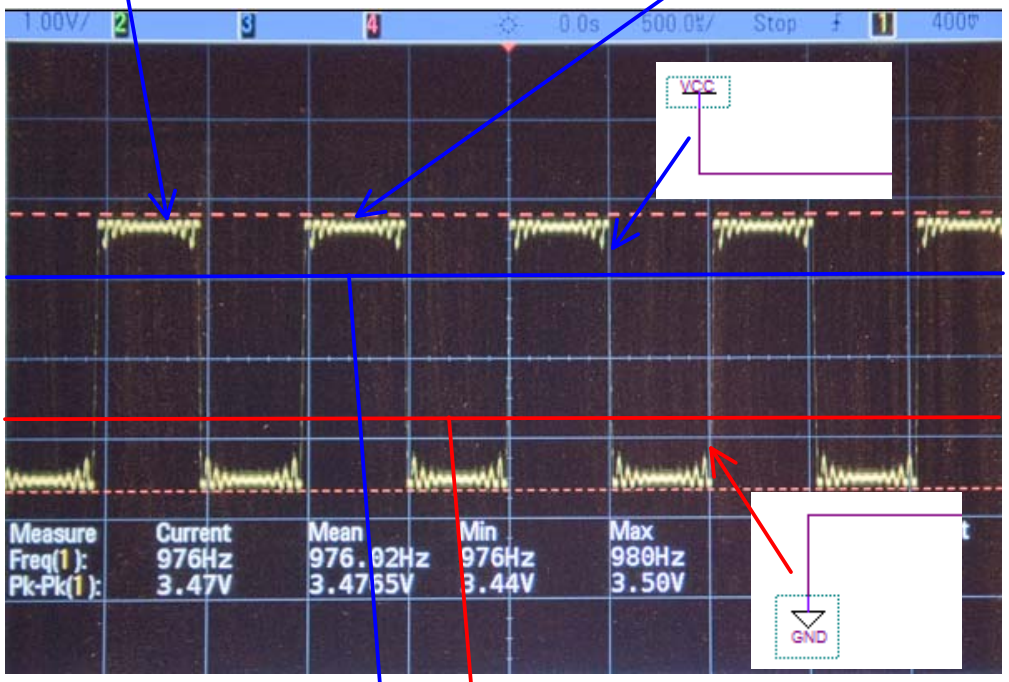
20 Pin -
Plastic Dual In-line Package (PDIP)

Mixed Signal Oscilloscope (MOS)

Analog (voltage) Signal

Actual Voltages

Analog



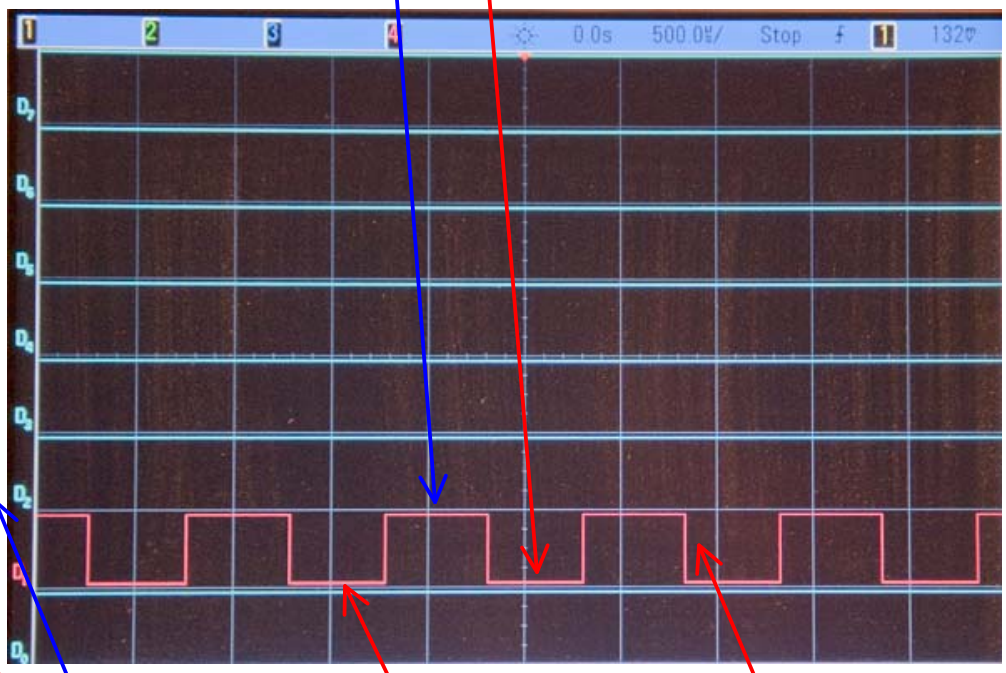
V_H

V_L

Interpretation

Digital signal is Hi or 1 if analog signal voltage > blue voltage

Digital signal is Lo or 0 if analog signal voltage < red voltage



Hi, 1

Lo, 0

Logical Values not actual Voltages

1 Bit (Binary Digit)

Digital Signal

Digital

Number Systems

Binary numbers
base, radix = 2

$A_{n-1} \quad A_1 A_0 \quad A_{-1} A_{-2} \quad A_{-m}$

$A_i \in \{0, 1\}$ $\quad A_i \rightarrow$ binary digit (bit)

Binary number

$\rightarrow 101101.1101 = (101101.1101)_2$

bit bit binary point

Converting from binary to decimal

$$(101101.11) = 1\times2^5 + 1\times2^4 + 0\times2^3 + 1\times2^2 + 0\times2^1 + 1\times2^0 + 1\times2^{-1} + 1\times2^{-2} = (53.75)_{10}$$

5 4 3 2 1 0 -1 -2

Notation: Code Composer Studio

$(101101)_2 \longrightarrow 101101b$ binary

msd (most significant digit) → 101101 ← lsd (least significant digit)

decimal integer to binary

$(41)_{10} = (?)_2$

$$\frac{41}{2} = 20 + \frac{1}{2}$$

$$\frac{20}{2} = 10 + \frac{0}{2}$$

$$\frac{10}{2} = 5 + \frac{0}{2}$$

$$\frac{5}{2} = 2 + \frac{1}{2}$$

$$\frac{2}{2} = 1 + \frac{0}{2}$$

$$\frac{1}{2} = 0 + \frac{1}{2}$$

$$(1\ 0\ 1\ 0\ 0\ 1)_2$$

Hexadecimal numbers

Hexadecimal radix = 16 $16 = 2^4$ radix is a power of 2

useful to represent binary numbers easily in compact form (easier on human beings than binary representation)

10 11 12 13 14 15

$\underbrace{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F}_{\text{16 hexadecimal digits}}$

$$(B65F)_{16} = 11 \times 16^3 + 6 \times 16^2 + 5 \times 16^0 + 15 \times 16^0$$
$$= (46687)_{10}$$

In Code Composer Studio:

zero

$(B65F)_{16} \longrightarrow$ 0XB65F (C style)

$\longrightarrow$ B65Fh (TI style)

Important table for converting bases among binary, octal and hexadecimal number representation

Decimal	Binary	Octal	Hex
00	0000	00	0
01	0001	01	1
02	0010	02	2
03	011	03	3
04	0100	04	4
05	0101	05	5
06	0110	06	6
07	0111	07	7
08	1000	10	8
09	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

Converting from binary to hexadecimal

$$(1011000110101 1)_2 = (?)_{16}$$

(use Table)

$$\underbrace{0010}_{2}\ \underbrace{1100}_{C}\ \underbrace{0110}_{6}\ \underbrace{1011}_{B}$$

pad zeros if required

start from here

$$= (2C6B)_{16}$$

Converting from hexadecimal to binary

$$(3A6)_{16} = (?)_2$$

(use Table)

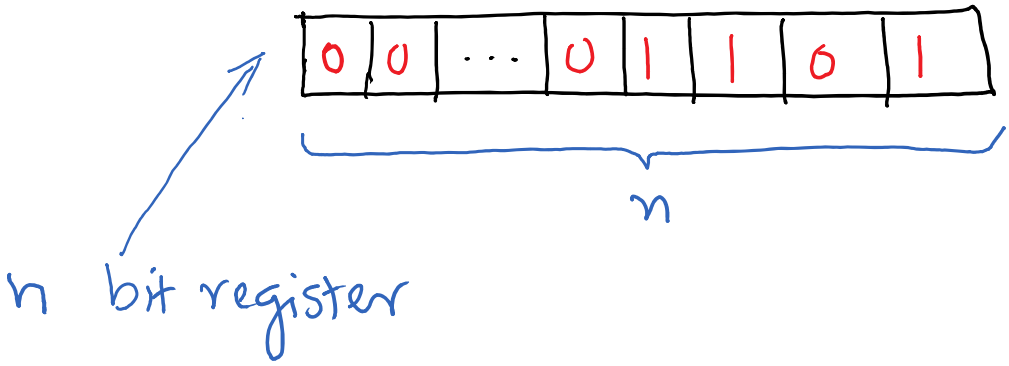
discard

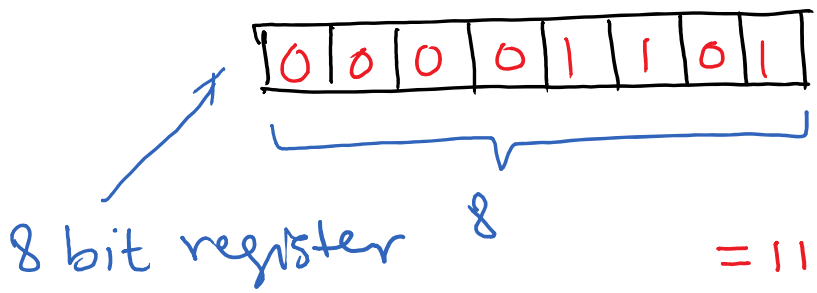
$$(3A6)_{16} = \underbrace{0011}\ \underbrace{1010}\ \underbrace{0110}$$

$$= (1110100110)_2$$

Compliments of Binary numbers.

Binary Number N = 1101 stored in an n bit register.





= 1101 stored in an 8 bit register

N = 1101

n = 8 (word length)

1's Compliment of Binary Numbers

register size

number

1's compliment of N $\rightarrow$ $(2^n - 1) - N$

Change all the 1's to 0, and all the 0's to 1 in N to get 1's compliment of N

$$10110010 \xrightarrow{\text{1's comp.}} 01001101$$

$$111 \xrightarrow{\text{1's comp}} 11111000$$

2's compliment of N $\rightarrow \begin{cases} 2^n - N & \text{for } N \neq 0 \\ 0 & \text{for } N = 0 \end{cases}$

2's comp. of N = 1's comp. of N + 1

$$101100 \xrightarrow{\text{2's comp}} 010011 \leftarrow \text{1's comp}$$

$$\begin{array}{r} 010011 \\ +\ 1 \\ \hline 010100 \end{array} \leftarrow \text{2's comp.}$$

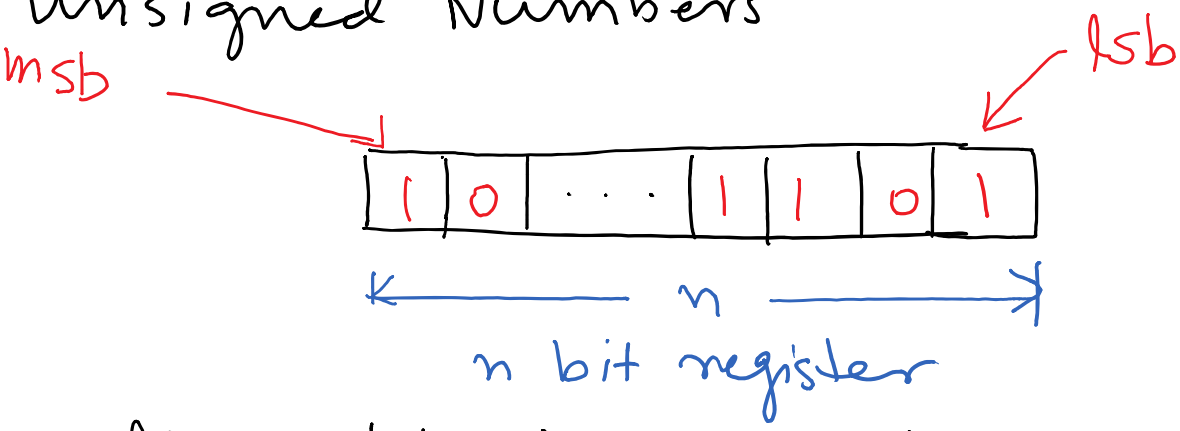
"Trick" to find 2's compliment of N:

Leave all the least significant zeros and the first 1 unchanged – replace all the remaining 0's by 1's, and 1's by 0's.

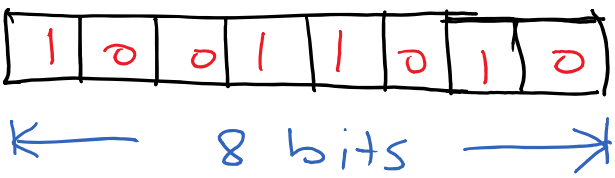
2's comp. of $\underbrace{1011}_{\text{invert these}}\underbrace{00}_{\text{leave these unchanged}} \longrightarrow 010100$

	n bits
2^n	1 0 0 0 … 0 0 0 0
2^n-1	1 1 1 … 1 1 1 1
2^{n-1}	1 0 0 … 0 0 0 0

Unsigned Numbers

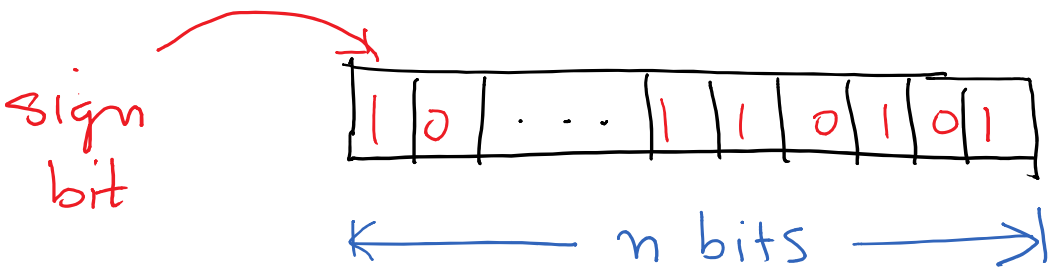


All the bits in the register represent the magnitude of the number.



Number $N = +(10011010)_2$

Signed Numbers
(2's compliment representation)



example 8 bit registers

$N = (9)_{10} \longrightarrow 0\,\underbrace{0000100\,1}_{\text{magnitude}}$

Sign bit
0 = +

magnitude
$= 0001001 = (9)_{10}$

$N = (-9)_{10} \longrightarrow$ 2's comp of 00001001

$= 11110111$

Sign bit
1 = −

Example

11101010

signed bit is 1 ∴ number is −ve

2's comp. of 11101010 is $00010110 = (22)_{10}$

∴ 11101010 is $(-22)_{10}$

00011011

signed bit 0 ∴ number is +ve

00011011 is $(27)_{10}$

Signed Binary Addition and subtraction

Addition

Add the two numbers (including the signed bit)
discard the carry out

$X = +6$	00000110
$Y = +13$	00001101
$Z = X + Y = +19$	0 0010011

signbit

$X = -6$	11111010
$Y = +13$	00001101
$Z = X + Y = +7$	1 00000111

discard signbit

$X = +6$	00000110
$Y = -13$	11110011
$Z = X + Y = -7$	11111001

signbit

2' comp of 11111001 is $\underbrace{0000\,0111}_{+}$ $= +7$

therefore 11111001 is -7

$X = -6$ 11111010

$Y = -13$ 11110011

$Z = X + Y = -19$ 1 11101101

discard sign bit

2' comp of 11101101 is 0 00100011 = +19

+ 19

therefore 11101101 is −19

Subtraction (2's comp. representation)

$Z = X - Y$

$Z = \{ X + (\text{2's comp of } Y)$

discard the carry out

$X = -6$ 11111010

$Y = -13 \xrightarrow{\text{2's comp of } -13}$ 00001101 } add

$Z = \underbrace{X - Y}_{-6-(-13)} = +7$ 1 00000111

discard sign bit

$X = +6$ 00000110

$Y = -13 \xrightarrow{\text{2's comp of } -13}$ 00001101 } add

$Z = \underbrace{X - Y}_{+6-(-13)} = +19$ 00010011

sign bit

overflow in 2's complement representation

Assume 8 bits

Sign bit

$0\underbrace{1111111}_{127} \longrightarrow +127$ ← maximum positive number

$11111111 \longrightarrow -1$

$10000001 \longrightarrow -127$

$10000000 \longrightarrow -128$ ← maximum negative number

Note: complementing the max. negative number → overflow
2's complement of 10000000 should give +128

but +128 needs more than 8 bits!

$\overbrace{0\,10000000}^{9\text{ bits}}$ +128
+

keep this "special case" in mind when converting the maximum negative number

8 bits can store a number if $-128 \leq \text{number} \leq +127$

otherwise there will be overflow

Range of 2's Comp

$$-2^{n-1} \leq N \leq 2^{n-1} - 1$$

Addition overflow in 2's compliment representation

8 bits 2's compliment representation
-128 <= N <= 127

Discard

addition

```
 10000001 = -127
 10000001 = -127
--------------
100000010  = 2  overflow!!
```

(2's compliment of 10000001 is 01111111 = 127
therefore 10000001 = -127)

```
  11000001     = -63
+ 11000001    =  -63                    -63+(-63) = -126
--------------
 110000010 =  -126 no overflow!!
```

(2's compliment of 10000010 is 01111110 = 126
therefore 10000010 = -126)

Carry out of signed bit position not equal to
Carry into signed bit position

```
 10
  10000001 = -127
+ 10000001 = -127
--------------
 100000010  = 2  overflow!!
```

Carry out of signed bit position is equal to
Carry into signed bit position

```
 11
  11000001 = -63
+ 11000001 = -63
--------------
 110000010  = -126  no overflow!!
```

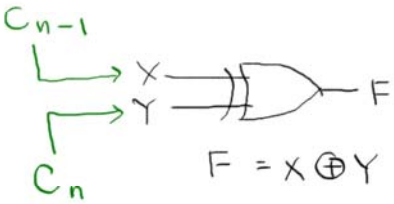
Discard

For addition (in 2's compliment representation),

- Overflow can only happen if both numbers have the same sign
- There is overflow if carry out of the signed bit position is not equal to carry into the signed bit position

overflow if

carry into sign bit position ≠ carry out of sign bit pos.



$F = X \oplus Y$

X	Y	F
0	0	0
0	1	1
1	0	1
1	1	0

→ overflow

1 0 0 0 0 0 0 0 → −128

2's complement = 1's complement + 1

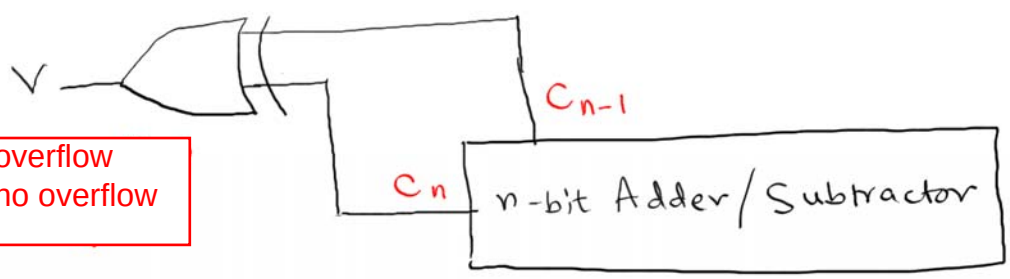
↑ → carry into signed bit position.

0 1 1 1 1 1 1 1 → 1's complement

+1

1 0 0 0 0 0 0 0

overflow



V = 1 - overflow
V = 0 - no overflow