Binary Review

Lecture 2
Section 2.4

Robb T. Koether

Hampden-Sydney College

Wed, Aug 28, 2019

Outline

Outline

Binary review

- When studying the processor at the hardware level we need to deal with binary numbers.
- Binary has two symbols: 0 and 1.
- A binary symbol is called a bit.

Binary review

- Bits are grouped to store more information.
  - 1 nibble = 4 bits.
  - 1 byte = 8 bits.
  - 1 halfword = 2 bytes = 16 bits.
  - 1 word = 4 bytes = 32 bits.

Binary review

- It is confusing to look at strings of 0s and 1s.

```
11111111111111111111111111111111
11111110000011111111000001111111
11111110000011111111000001111111
11111110000011111111000001111111
11111111111111111111111111111111
11111111111111111111111111111111
11110001111111111111111110001111
11110000000111111111100000001111
11111110000000000000000001111111
11111111111000000000011111111111
11111111111111111111111111111111
```

- The problem is too many of the same symbol.

Binary review

- Let one symbol represent a group of bits.
  - 1-bit can represent 2 different things.
  - 2-bits can represent 4 different things.
  - 3-bits can represent 8 different things.
  - 4-bits can represent 16 different things.
- In general, n-bits can represent 2^n different things.

Binary review

- Four bits (half a byte) represent sixteen things.
- Hexadecimal consists of 16 different symbols.
- 0, . . . , 9, A, B, C, D, E, F.

Binary	Hex Symbol	Binary	Hex Symbol
0000	0	1000	8
0001	1	1001	9
0010	2	1010	A
0011	3	1011	B
0100	4	1100	C
0101	5	1101	D
0110	6	1110	E
0111	7	1111	F

Binary review

- To convert from binary to hexadecimal
  - Group bits in fours from the right end of the string. (Add leading 0s if necessary.)
  - Substitute the corresponding hexadecimal digit.

$$\begin{aligned} 1111101011001110_2 &= 1111_2 \mid 1010_2 \mid 1100_2 \mid 1110_2 \\ &= \mathrm{F} \mid \mathrm{A} \mid \mathrm{C} \mid \mathrm{E} \\ &= \mathrm{FACE}_{16}. \end{aligned}$$

Binary review

- To convert from hexadecimal to binary, reverse the process.

$$\begin{aligned} \text{FACE}_{16} &= \text{F} \mid \text{A} \mid \text{C} \mid \text{E} \\ &= 1111_2 \mid 1010_2 \mid 1100_2 \mid 1110_2 \\ &= 1111101011001110_2 \end{aligned}$$

Outline

Storing Negative Integers

- To store negative integers, we divide the range 0 to $2^{32} - 1$ in half.
  - The lower half 0 to $2^{31} - 1$ (hex `00000000` to hex `7FFFFFFF`) represents the positive integers 0 to $2^{31} - 1$.
  - The upper half 2^{31} to $2^{32} - 1$ (hex `80000000` to hex `FFFFFFFF`) represents the negative integers.
- Negative integers are interpreted in two's complement notation.

3-Bit Signed and Unsigned Integers

Unsigned	
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

Signed	
000	0
001	1
010	2
011	3
100	−4
101	−3
110	−2
111	−1

3-bit integers, unsigned and signed

3-Bit Unsigned Integers

Unsigned	
00000000000000000000000000000000	0
00000000000000000000000000000001	1
00000000000000000000000000000010	2
:	:
01111111111111111111111111111111	$2^{31} - 1$
10000000000000000000000000000000	2^{31}
10000000000000000000000000000001	$2^{31} + 1$
10000000000000000000000000000010	$2^{31} + 2$
:	:
11111111111111111111111111111111	$2^{32} - 1$

32-bit unsigned integers

3-Bit Signed Integers

Signed	
00000000000000000000000000000000	0
00000000000000000000000000000001	1
00000000000000000000000000000010	2
:	:
01111111111111111111111111111111	$2^{31} - 1$
10000000000000000000000000000000	-2^{31}
10000000000000000000000000000001	$-2^{31} + 1$
10000000000000000000000000000010	$-2^{31} + 2$
:	:
11111111111111111111111111111111	-1

32-bit signed integers

3-Bit Signed and Unsigned Integers

Unsigned	
00000000000000000000000000000000	0
00000000000000000000000000000001	1
00000000000000000000000000000010	2
:	:
01111111111111111111111111111111	$2^{31} - 1$
10000000000000000000000000000000	2^{31}
10000000000000000000000000000001	$2^{31} + 1$
10000000000000000000000000000010	$2^{31} + 2$
:	:
11111111111111111111111111111111	$2^{32} - 1$

Signed	
00000000000000000000000000000000	0
00000000000000000000000000000001	1
00000000000000000000000000000010	2
:	:
01111111111111111111111111111111	$2^{31} - 1$
10000000000000000000000000000000	-2^{31}
10000000000000000000000000000001	$-2^{31} + 1$
10000000000000000000000000000010	$-2^{31} + 2$
:	:
11111111111111111111111111111111	-1

32-bit integers, unsigned and signed

Two's Complement

- Negative integers are interpreted in two's complement notation.
- The `1` in bit 31 (MSB) indicates a negative integer.

3-Bit Signed and Unsigned Integers

Unsigned	
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

Signed	
000	0
001	1
010	2
011	3
100	−4
101	−3
110	−2
111	−1

3-bit integers, unsigned and signed

Two's Complement Example

Example (Two's Complement Example)

- The absolute value of the negative integer (i.e., the corresponding positive integer) is obtained by reversing all the bits and then adding `1`.

Two's Complement Example

Example (Two's Complement Example)

- The absolute value of the negative integer (i.e., the corresponding positive integer) is obtained by reversing all the bits and then adding `1`.
- What negative number is represented by

  `1111 1111 1111 1111 1111 0000 1100 1110`?

Two's Complement Example

Example (Two's Complement Example)

- The absolute value of the negative integer (i.e., the corresponding positive integer) is obtained by reversing all the bits and then adding `1`.
- What negative number is represented by

  `1111 1111 1111 1111 1111 0000 1100 1110`?
- Reverse all the bits:

  `0000 0000 0000 0000 0000 1111 0011 0001`

Two's Complement Example

Example (Two's Complement Example)

- The absolute value of the negative integer (i.e., the corresponding positive integer) is obtained by reversing all the bits and then adding `1`.
- What negative number is represented by

  `1111 1111 1111 1111 1111 0000 1100 1110`?
- Reverse all the bits:

  `0000 0000 0000 0000 0000 1111 0011 0001`
- Add `1`:

  `0000 0000 0000 0000 0000 1111 0011 0010`

Two's Complement Example

Example (Two's Complement Example)

- The absolute value of the negative integer (i.e., the corresponding positive integer) is obtained by reversing all the bits and then adding `1`.
- What negative number is represented by

  `1111 1111 1111 1111 1111 0000 1100 1110`?
- Reverse all the bits:

  `0000 0000 0000 0000 0000 1111 0011 0001`
- Add `1`:

  `0000 0000 0000 0000 0000 1111 0011 0010`
- The value is -3890.

Outline

Binary Arithmetic

- Perform the following additions as 8-bit unsigned integers;
  - $00001011 + 00000110$
  - $11110001 + 00000101$
  - $11110001 + 11111101$
- In which case(s) was there overflow?
- How is overflow detected?

Binary Arithmetic

- Perform the following additions as 8-bit signed integers;
  - $00001011 + 00000110$
  - $01111011 + 01110110$
  - $11110001 + 00000101$
  - $11110001 + 11111101$
  - $10000111 + 10001101$
- In which case(s) was there overflow?
- How is overflow detected?

Binary Arithmetic

- Subtraction is performed by *adding* the two's complement of the subtrahend.
- Perform the following subtractions as 8-bit unsigned integers;
  - `00001111` − `00000110`
  - `00000011` − `00000110`
- In which case(s) was there overflow?
- How is overflow detected?

Binary Arithmetic

- Perform the following subtractions as 8-bit signed integers;
  - `00001111 − 00000110`
  - `00000011 − 00000110`
  - `01111100 − 11110110`
  - `10000101 − 00001110`
- In which case(s) was there overflow?
- How is overflow detected?

Outline

Outline

Unsigned Addition Overflow

x MSB	y MSB	$x + y$ MSB	Overflow?
0	0	0	No
0	0	1	No
0	1	0	Yes
0	1	1	No
1	0	0	Yes
1	0	1	No
1	1	0	Yes
1	1	1	Yes

Unsigned Addition Overflow

x MSB	y MSB	$x + y$ MSB	Overflow?
0	0	0	No
0	0	1	No
0	1	0	Yes
0	1	1	No
1	0	0	Yes
1	0	1	No
1	1	0	Yes
1	1	1	Yes

Outline

Unsigned Subtraction Overflow

x MSB	y MSB	$x - y$ MSB	Overflow?
0	0	0	No
0	0	1	Yes
0	1	0	Yes
0	1	1	Yes
1	0	0	No
1	0	1	No
1	1	0	No
1	1	1	Yes

Unsigned Subtraction Overflow

x MSB	y MSB	$x - y$ MSB	Overflow?
0	0	0	No
0	0	1	Yes
0	1	0	Yes
0	1	1	Yes
1	0	0	No
1	0	1	No
1	1	0	No
1	1	1	Yes

Outline

Signed Addition Overflow

x MSB	y MSB	$x + y$ MSB	Overflow?
0	0	0	No
0	0	1	Yes
0	1	0	No
0	1	1	No
1	0	0	No
1	0	1	No
1	1	0	Yes
1	1	1	No

Signed Addition Overflow

x MSB	y MSB	$x + y$ MSB	Overflow?
0	0	0	No
0	0	1	Yes
0	1	0	No
0	1	1	No
1	0	0	No
1	0	1	No
1	1	0	Yes
1	1	1	No

Outline

Signed Subtraction Overflow

x MSB	y MSB	$x - y$ MSB	Overflow?
0	0	0	No
0	0	1	No
0	1	0	No
0	1	1	Yes
1	0	0	Yes
1	0	1	No
1	1	0	No
1	1	1	No

Signed Subtraction Overflow

x MSB	y MSB	$x - y$ MSB	Overflow?
0	0	0	No
0	0	1	No
0	1	0	No
0	1	1	Yes
1	0	0	Yes
1	0	1	No
1	1	0	No
1	1	1	No

Outline

Assignment

Assignment

- Read Section 2.4.