CSSE2010 / CSSE7201
Learning Lab 1

Number Representations

School of Information Technology and Electrical Engineering
The University of Queensland

CSSE2010/7201 - Learning Lab 1
Number Representations

- Make sure you have
  - Something to write on and with
  - Your response device for polls (app or web)
  - For app, select ‘East Asia’ and the web URL is http://responsewaresg.net/ ✓ Session ID: CSSE2010EXT
- You may find a calculator useful if you have one (or use calculator on PC)

CSSE2010

What happens in Learning Lab sessions?

A mix of:

- Mini-lectures (lots of content today, won't usually be this much)
- Problem Solving: Simulations, building up circuits on a breadboard, designing and writing software (C/Assembly), and debugging
- Poll questions
  - **Some are individual** to test **your** understanding and let us know how many people are on the right track
  - Copying someone else does not help you or us
- Prac activities (hands-on with equipment/software)
  - Not today – this starts next week
- **Today: Kit distribution to internal (IN) students.**
- **EX students: Your labs for the first half of the course will be based on Logisim software. For the second half (i.e. from week 6-7) you will need the hardware items that you are supposed to acquire. A list is given in the ECP.**

Today

Number Representations

- Octal, hexadecimal, binary
- Negative Numbers
  - Signed magnitude
  - Ones' complement
  - Two's complement
  - Excess 2^{m-1}

CSSE2010

Poll question: What is 10010010 converted to decimal?

- 0% A. -110
- 0% B. -109
- 0% C. -18
- 3% D. 18
- 0% E. 82
- 87% F. 146
- 5% G. None of the above
- 5% H. I don’t know

$$2^7\ 2^6\ 2^5\ 2^4\ 2^3\ 2^2\ 2^1\ 2^0$$
$$1\ 0\ 0\ 1\ 0\ 0\ 1\ 0$$
$$2^7 + 2^4 + 2^1 = 128 + 16 + 2$$
$$= 146$$

URL: http://responsewaresg.net/

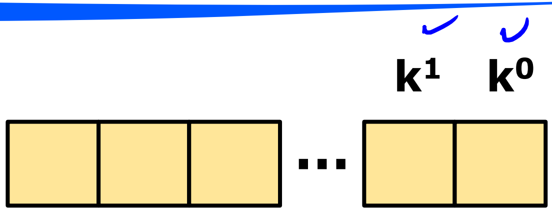
Session ID: CSSE2010EXT

0

Radices (from lecture 1)

base

- A radix-k number system   radix-10   10 symbols   0, 1, 2.... 9
  - k different symbols to represent digits 0 to k-1
  - Value of each digit is (from the right) k^0, k^1, k^2, k^3, …
- Often convenient to deal with
  - **Octal** (radix-8)
    - Symbols: 0 1 2 3 4 5 6 7
    - One octal digit corresponds to 3 bits
  - **Hexadecimal** (radix-16)   10 11 12 13 14 15
    - Symbols: 0 1 2 3 4 5 6 7 8 9 A B C D E F
    - One hexadecimal digit corresponds to 4 bits (useful!)



N-digit radix-k number

byte = 8-bi

← 2 hex

Dec	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Oct	0	1	2	3	4	5	6	7	10	11	12	13	14	15	16	17	20	21
Hex	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11

CSSE2010

Poll question – What's the maximum value you can represent in n radix-k digits?

responseware sg.net

Session ID: CSSE2010EXT

0%	A.	nk
11%	B.	$nk-1$
0%	C.	k^n
32%	D.	$(k^n)-1$
35%	E.	$k^{(n-1)}$
8%	F.	$(n^k)-1$
8%	G.	nk^n
5%	H.	$(nk^n)-1$

Tip: consider examples in base 10 or 2

2-digit base 10 → 99 → $10^2 - 1 = 99$

3-digit " → 999 → $10^3 - 1 = 999$

$k^n - 1$

4-digit base 2

1111 → 15 → $2^4 - 1 = 15$

Answer Individually (No discussion)

This is to test your individual understanding.

There will be discussion afterwards (unless all correct)

0

Conversions

$2^2\ 2^1\ 2^0$

$101 \rightarrow 2^2 + 2^0 = 4+1 = 5$

1 • Convert 1001101001 (unsigned binary) to
  - octal ✓
  - hexadecimal (hex) ✓

$2^3 = 8$

1 1 5 1

001 | 001 | 101 | 001

$2^4 = 16$

2 6 9

0010 | 0110 | 1001

2 • Convert C55E7201 (hex) to
  - octal ✓

0 … 9
A B C D E F
10 11 12

hex → binary → octal

3 0 5 2 7 4 7 1 0 0 1

0 11 | 00 0 | 101 | 0101 | 1110 | 0111 | 0010 | 0000 | 0001

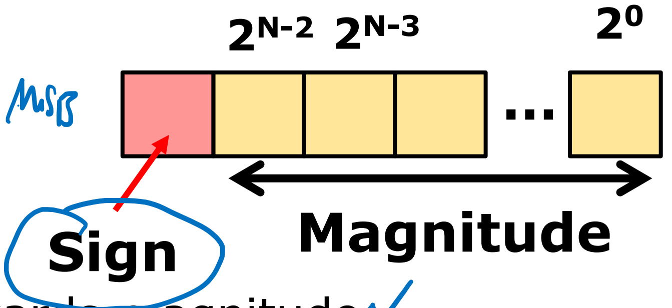
C 5 5 E 7 2 0 1

Negative Numbers

- Computers don’t store + and – signs, must use binary digits (0,1)
- 4 different formats have been used…
  - Signed magnitude ✓
  - Ones’ complement ✓
  - **Two’s complement – this is what is practically used. We’ll see why** ✓
  - Excess 2^{m-1} ✓

Negative Numbers - Signed Magnitude Representation

- **Signed magnitude**
  - Leftmost bit = **sign-bit**
    - **0** = positive ✓
    - **1** = negative ✓
  - Negate by
    - Inverting sign-bit
  - Sign bit does not contribute towards magnitude ✓
  - Example: 8-bit signed magnitude representations of +47 and -47



$2^6\ 2^5\ 2^4\ 2^3\ 2^2\ 2^1\ 2^0$

+47: 0 0 1 0 1 1 1 1

-47: 1 0 1 0 1 1 1 1

-ve

What range of numbers can be represented with 8-bit signed-magnitude binary?

- 3% A. 0 to 255
- 9% B. -255 to 255
- 9% C. -256 to 256
- 21% D. -128 to 127
- 3% E. -128 to 128
- ✓ 45% F. -127 to 127 ✓✓
- 3% G. -127 to 128
- 6% H. None of the above

1-bit for sign ✓
7-bits for magnitudes ✓

$k^n - 1$

$2^7 - 1 = 127$

Largest -ve number:
sign bit is -ve
Largest magnitude

Largest +ve number
sign bit +ve
Largest magnitude = 127

Answer Individually (No discussion initially)

0

CSSE2010

What range of numbers can be represented with n-bit signed-magnitude binary?

0%	A.	0 to 2^n-1
35%	B.	$-(2^n-1)$ to 2^n-1 ×
0%	C.	-2^n to 2^n
0%	D.	$-(2^{(n-1)}-1)$ to $2^{(n-1)}$
6%	E.	$-2^{(n-1)}$ to $2^{(n-1)}$
59%	F.	$-(2^{(n-1)}-1)$ to $2^{(n-1)}-1$ ✓
0%	G.	$-2^{(n-1)}$ to $2^{(n-1)}-1$
0%	H.	None of the above

n = 8-bit → 7 bits for magnitude

$-(2^7-1)$ to 2^7-1

-127 to 127

$-(2^{n-1}-1)$ to $2^{n-1}-1$

0

Negative Numbers – Ones' complement

- **Ones' complement**
  - Leftmost bit = sign-bit (as per signed magnitude, **but now also contributes towards magnitude in negative**)
    - 0 = positive
    - 1 = negative
  - Bit position values changed as →

$-(2^{N-1}-1)$	2^{N-2}	2^{N-3}	... 2^1	2^0
			...	

  - By having the bit position values like this allows us to find the negative number by just **inverting all the bits**
  - So, if you want to find (–A), get the binary representation of A and **obtain its ones' complement**, i.e. by inverting all the bits
  - See, a simple 4-bit example: ones' complement representation of +3 and -3

+3: 0011

$-(2^3-1)$ 2^2 2^1 2^0

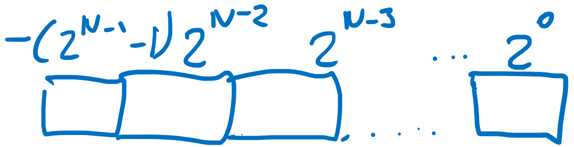
→ 4 2 1

-3: 1100

→ 7 4 0 0 = -3

What range of numbers can be represented with 8-bit ones' complement binary?

- 4% A. 0 to 255
- 19% B. -255 to 255
- 0% C. -256 to 256
- 31% D. -128 to 127
- 12% E. -128 to 128
- ✓ 35% F. -127 to 127
- 0% G. -127 to 128
- 0% H. None of the above



Largest -ve number

1000 0000 $-(2^7-1) = -127$ ✓

Largest +ve number

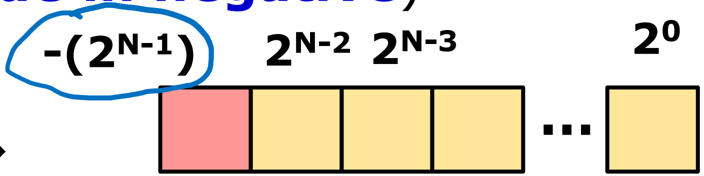
0111 1111 $= 2^7-1 = 127$

Answer Individually (No discussion initially)

0

Negative Numbers – Two’s complement

- **Two’s complement**
  - Leftmost bit = sign-bit (as per signed magnitude, **but now also contributes towards magnitude in negative**)
    - 0 = positive
    - 1 = negative
  - Bit position values changed as →



  - By having the bit position values like this allows us to find the negative number by **inverting all the bits and adding 1**
  - So, if you want to find (–A), get the binary representation of A and **obtain its two’s complement**, i.e. inverting all the bits and add 1
  - See, a simple 4-bit example: ~~ones’~~ two complement representation of +3 and -3

+3: 0011 → -3: 1100
+ 1
1101

$-2^3 \; 2^2 \; 2^1 \; 2^0$
$-8 \; 4 \; 2 \; 1$

$-8 + 4 + 1 = -3$

Comparing signed magnitude with two's complement

- Signed magnitude (8-bit)

Sign bit

	6	5	4	3	2	1	0
0: +	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1: -	64	32	16	8	4	2	1

- Two's complement format (8-bit)

7	6	5	4	3	2	1	0
-2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
-128	64	32	16	8	4	2	1

CSSE2010

What range of numbers can be represented with 8-bit two's complement binary?

- 0% A. 0 to 255
- 7% B. -255 to 255
- 0% C. -256 to 256
- ✓ 45% D. -128 to 127 ✓✓
- 24% E. -128 to 128
- 14% F. -127 to 127
- 10% G. -127 to 128
- 0% H. None of the above

$-(2^{N-1})\quad 2^{N-2}\quad 2^{N-3}\quad \cdots\quad 2^{0}$

Largest -ve number

$1000\ 0000 = -2^{8-1} = -128$

Largest +ve number

$0111\ 1111 = 2^{7}-1 = 127$

0

Answer Individually (No discussion initially)

What range of numbers can be represented with n-bit two's complement binary?

0% A. 0 to 2^n-1

0% B. $-(2^n-1)$ to 2^n-1

0% C. -2^n to 2^n

0% D. $-(2^{(n-1)}-1)$ to $2^{(n-1)}$

3% E. $-2^{(n-1)}$ to $2^{(n-1)}$

3% F. $-(2^{(n-1)}-1)$ to $2^{(n-1)}-1$

✓ 87% G. $-2^{(n-1)}$ to $2^{(n-1)}-1$ ✓

0% H. None of the above

$-(2^{N-1})\ 2^{N-2}\ 2^{N-3}\ \ldots\ 2^0$

$-ve: -2^{N-1}$

$+ve: 2^{N-1}-1$

0

Negative Numbers – Excess 2^{m-1} ✓

0 sign-bit = -ve
1 sign-bit = +ve

- **Excess 2^{m-1}**
  - e.g for 8 bits, "excess 128" ✓
  - Number stored as true value plus 128 ✓
    - e.g. -3 stored as -3 + 128 = 125 ✓
    - e.g. 3 stored as 3 + 128 = 131 ✓
  - Interestingly, same as two's complement with sign-bit reversed (0 in most significant bit means bit value of -128, 1 means bit value of 0)
  - Practical use: hardware becomes simple for comparing two signed numbers. ✓

    ↳ essentially comparing unsigned numbers
  - Example: Excess-128 representations of +47 and -47

+47 => 175
10101111

-47 => 81
01010001

10101 × } unsigned
larger → 11010 ✓

What range of numbers can be represented with 8-bit excess-128 binary?

- 26% A. 0 to 255
- 4% B. -255 to 255
- 4% C. -256 to 256
- 39% D. -128 to 127 ✓
- 13% E. -128 to 128
- 0% F. -127 to 127
- 9% G. -127 to 128
- 4% H. None of the above

-ve → negative
+ve → positive

Largest -ve number

0000 0000 = 0 − 128 = −128

Largest +ve

1111 1111

$2^8 - 1 = 255 - 128 = 127$

0

Answer Individually (No discussion initially)

Limitations/Advantages

- Two representations of zero - undesirable
  - Signed-magnitude ✓
    - e.g. for 8 bits: 00000000, 10000000 (+0, −0)
  - Ones' complement ✓
    - e.g. for 8 bits: 00000000, 11111111 (+0, −0)
  - This means we've wasted one bit pattern which could've been used to increase the range. ✓✓
- Asymmetric range
  - Two's complement
  - Excess 2^{m-1}

8-bits: −128 − 127

2's comp +0: 0000 0000
−0: 1111 1111
+ 1
0 000 0000 = +0

Excess: +0 -> 0+128 = 128 = 1000 0000, −0 -> −0 +128 = 128 = 1000 0000 = +0

Poll question: What is 10010010 converted to decimal?

0% A. -110

0% B. -109

6% C. -18

0% D. 18

0% E. 82

X 39% F. 146 Unsigned

11% G. None of the above

44% H. I don't know

0

Format matters!

- Binary data is meaningless unless you know the format
- Example: **01010101** could be
  - 85 (if format is an 8-bit unsigned number) ✓
  - -43 (if format is excess-128) ✓
  - 'U' (if format is an ASCII character) ✓
  - 10.625 (if format is an 8-bit unsigned fixed-point number with 5 bits before the binary point and 3 after) ✓
- The format is not encoded in the data, this must be known separately, by e.g.
  - A human ✓
  - A computer program (software) unsigned int, int, char
  - Digital hardware (e.g. CPU) ✓

Complete the following by yourself and observe changes across various formats

4-bit unsigned

0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	10
1011	11
1100	12
1101	13
1110	14
1111	15

4-bit sign-mag

0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	-0
1001	-1
1010	-2
1011	-3
1100	-4
1101	-5
1110	-6
1111	-7

4-bit 1's comp

0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	-7
1001	-6
1010	-5
1011	-4
1100	-3
1101	-2
1110	-1
1111	-0

4-bit 2's comp

0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	-8
1001	-7
1010	-6
1011	-5
1100	-4
1101	-3
1110	-2
1111	-1

4-bit excess-8

0000	-8
0001	-7
0010	-6
0011	-5
0100	-4
0101	-3
0110	-2
0111	-1
1000	0
1001	1
1010	2
1011	3
1100	4
1101	5
1110	6
1111	7

Kit Borrowing for IN Students

- Sign and submit the kit borrowing agreement and collect your kit. Wait in your tables, do not queue up.
- Check if you have all the parts
- Do not use the kit until we advise to do so – there are things that you can easily break
- For next week, you will use the logic chips and breadboard and will be advised how to use them.
- The AVR microcontroller is needed only in the second half of the course
- IN students – if you either drop the course or change to EX mode after borrowing the kit, you must return the kit within 3 business days by contacting the course coordinator via email.

EX students: you don't need hardware until week 6. But start acquiring your hardware items now (the list is on the ECP). Ask questions if you are unsure about any part.