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/
- You may find a calculator useful if you have one (or use calculator on PC)

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

Poll question: What is 10010010 converted to decimal?

13%	A.	-110
13%	B.	-109
13%	C.	-18
13%	D.	18
13%	E.	82
13%	F.	146
13%	G.	None of the above
13%	H.	I don't know

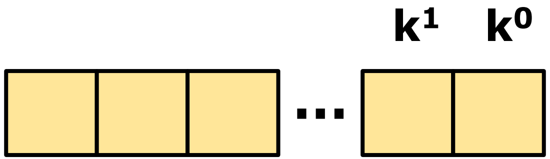
URL: http://responsewaresg.net/

Session ID:

10

Radices (from lecture 1)

- A radix-k number system
  - 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)
    - 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

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?

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

Answer Individually (No discussion)
This is to test your individual understanding.
There will be discussion afterwards (unless all correct)

10

Conversions

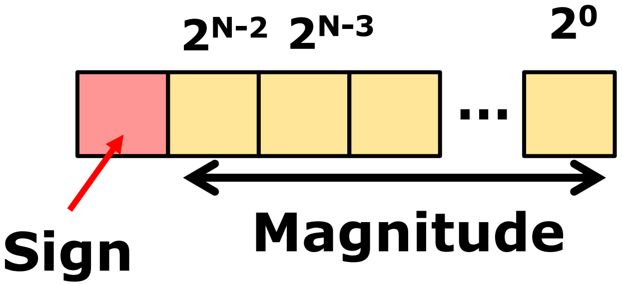
- Convert 1001101001 (unsigned binary) to
  - octal
  - hexadecimal (hex)
- Convert C55E7201 (hex) to
  - octal

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



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

- 0% A. 0 to 255
- 0% B. -255 to 255
- 0% C. -256 to 256
- 0% D. -128 to 127
- 0% E. -128 to 128
- 0% F. -127 to 127
- 0% G. -127 to 128
- 0% H. None of the above

Answer Individually (No discussion initially)

10

What range of numbers can be represented with n-bit signed-magnitude 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 $2^{(n-1)}$

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

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

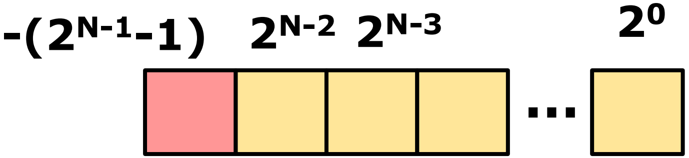
0% G. -$2^{(n-1)}$ to $2^{(n-1)}$-1

0% H. None of the above

10

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 →



  - 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

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

0% A. 0 to 255

0% B. -255 to 255

0% C. -256 to 256

0% D. -128 to 127

0% E. -128 to 128

0% F. -127 to 127

0% G. -127 to 128

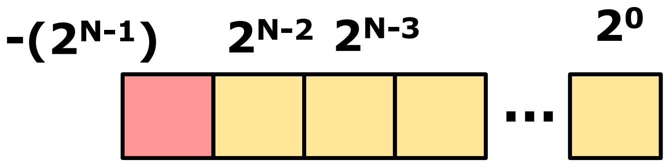
0% H. None of the above

Answer Individually (No discussion initially)

10

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' complement representation of +3 and -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

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

0% A. 0 to 255
0% B. -255 to 255
0% C. -256 to 256
0% D. -128 to 127
0% E. -128 to 128
0% F. -127 to 127
0% G. -127 to 128
0% H. None of the above

10

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 $2^{(n-1)}$

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

0% 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

10

Negative Numbers – Excess 2^{m-1}

- **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.
  - Example: Excess-128 representations of +47 and -47

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

0% A. 0 to 255

0% B. -255 to 255

0% C. -256 to 256

0% D. -128 to 127

0% E. -128 to 128

0% F. -127 to 127

0% G. -127 to 128

0% H. None of the above

10

Answer Individually (No discussion initially)

Limitations/Advantages

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

Poll question: What is 10010010 converted to decimal?

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

10

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)
  - Digital hardware (e.g. CPU)

Complete the following by yourself and observe changes across various formats

4-bit unsigned	4-bit sign-mag	4-bit 1's comp	4-bit 2's comp	4-bit excess-8
0000	0000	0000	0000	0000
0001	0001	0001	0001	0001
0010	0010	0010	0010	0010
0011	0011	0011	0011	0011
0100	0100	0100	0100	0100
0101	0101	0101	0101	0101
0110	0110	0110	0110	0110
0111	0111	0111	0111	0111
1000	1000	1000	1000	1000
1001	1001	1001	1001	1001
1010	1010	1010	1010	1010
1011	1011	1011	1011	1011
1100	1100	1100	1100	1100
1101	1101	1101	1101	1101
1110	1110	1110	1110	1110
1111	1111	1111	1111	1111

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.