CSc 106: The Practice of Computer Science

Binary, Hexadecimal and 2s Complement and Algorithms continued…

Question 1

- What is the hexadecimal value of the following subtraction of 2's complement values:
  $2F_{16} - CF_{16}$

A. $A0_{16}$

B. 80_{16}

C. $5F_{16}$

D. 60_{16}

E. $9F_{16}$

Thinking in hex…

- Recall: we can write the binary codes for the number 0-15 as shown below.
- Write the corresponding hexadecimal value in the space provide below each

0000	0001	0010	0011	0100	0101	0110	0111
1000	1001	1010	1011	1100	1101	1110	1111

Thinking in hex…

- Recall: we can write the binary codes for the number 0-15 as shown below.
- Write the corresponding hexadecimal value in the space provided below each

0000	0001	0010	0011	0100	0101	0110	0111
0x0	**0x1**	**0x2**	**0x3**	**0x4**	**0x5**	**0x6**	**0x7**
1000	1001	1010	1011	1100	1101	1110	1111
0x8	**0x9**	**0xa**	**0xb**	**0xc**	**0xd**	**0xe**	**0xf**

Question 5a

- convert the binary number
  1000 1111 1001 1101 to hexadecimal
  A. 0xafbd
  B. 0x8e9c
  C. 0x815913
  D. 0x8f9d

Question 5b

- convert the hexadecimal number 0xa09d to binary

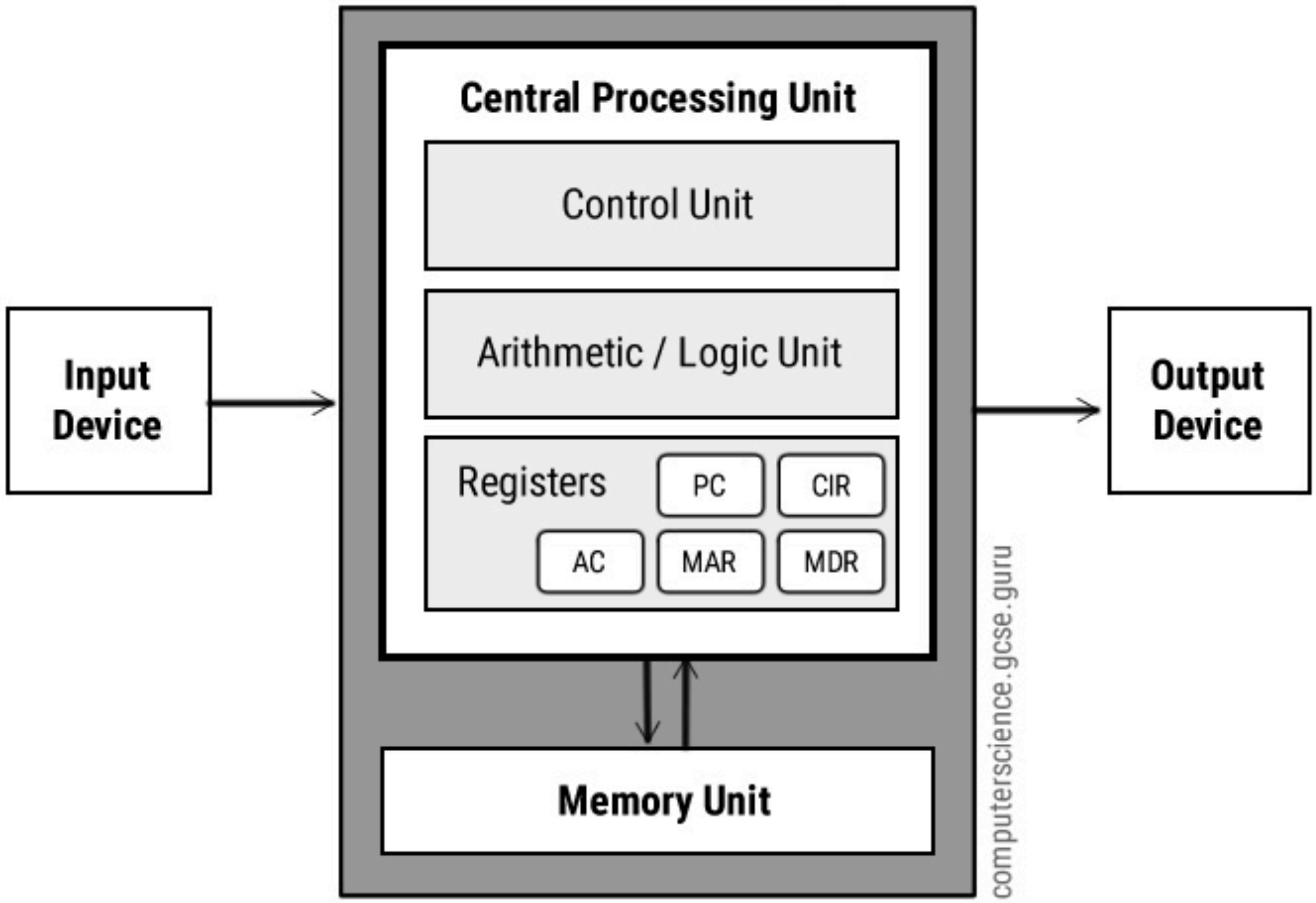
A. 100913

B. 1000 0000 1001 1110

C. 1010 0000 1001 1101

D. 1011 0000 1001 1110

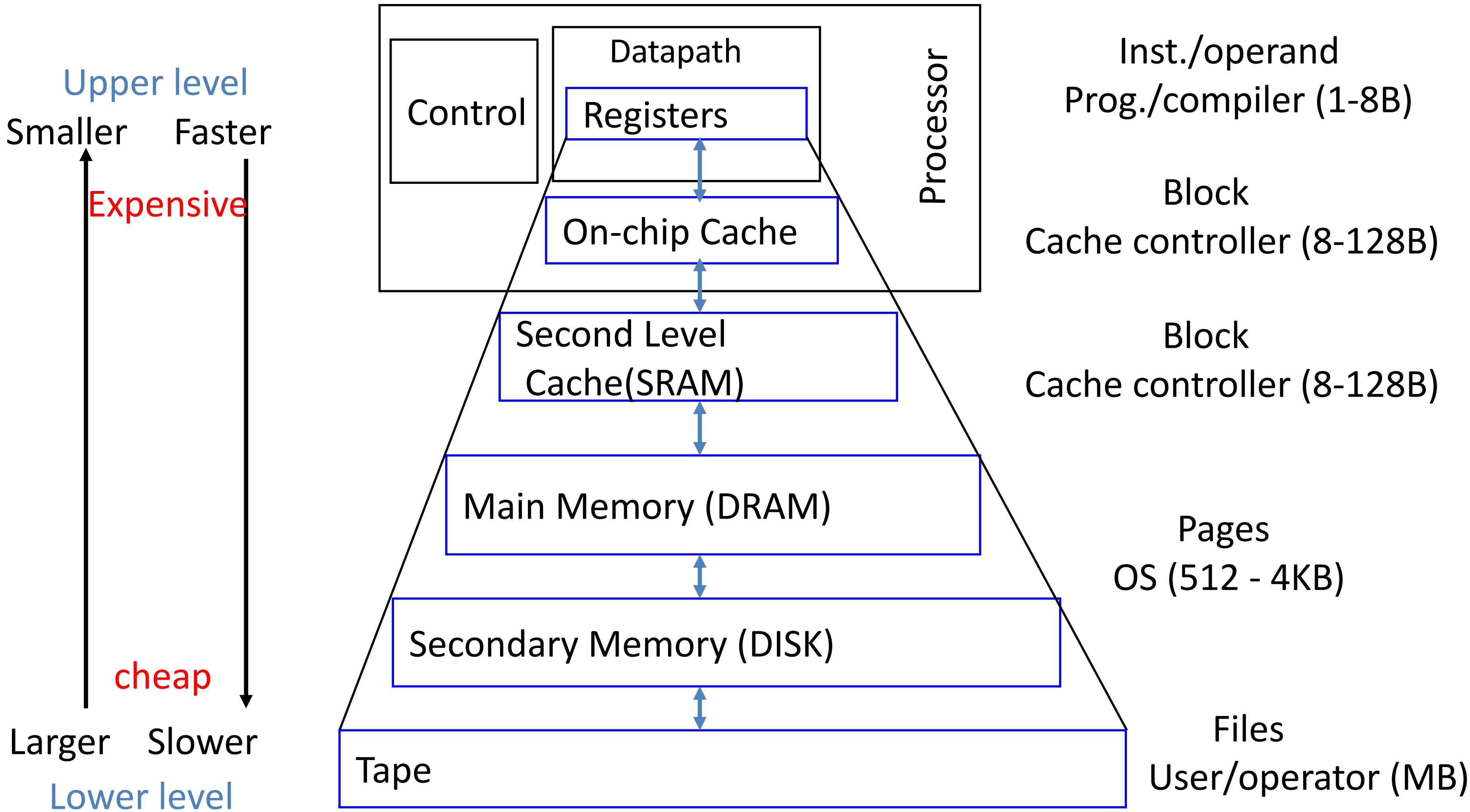
von Neuman Architecture



https://www.computerscience.gcse.guru/theory/von-neumann-architecture

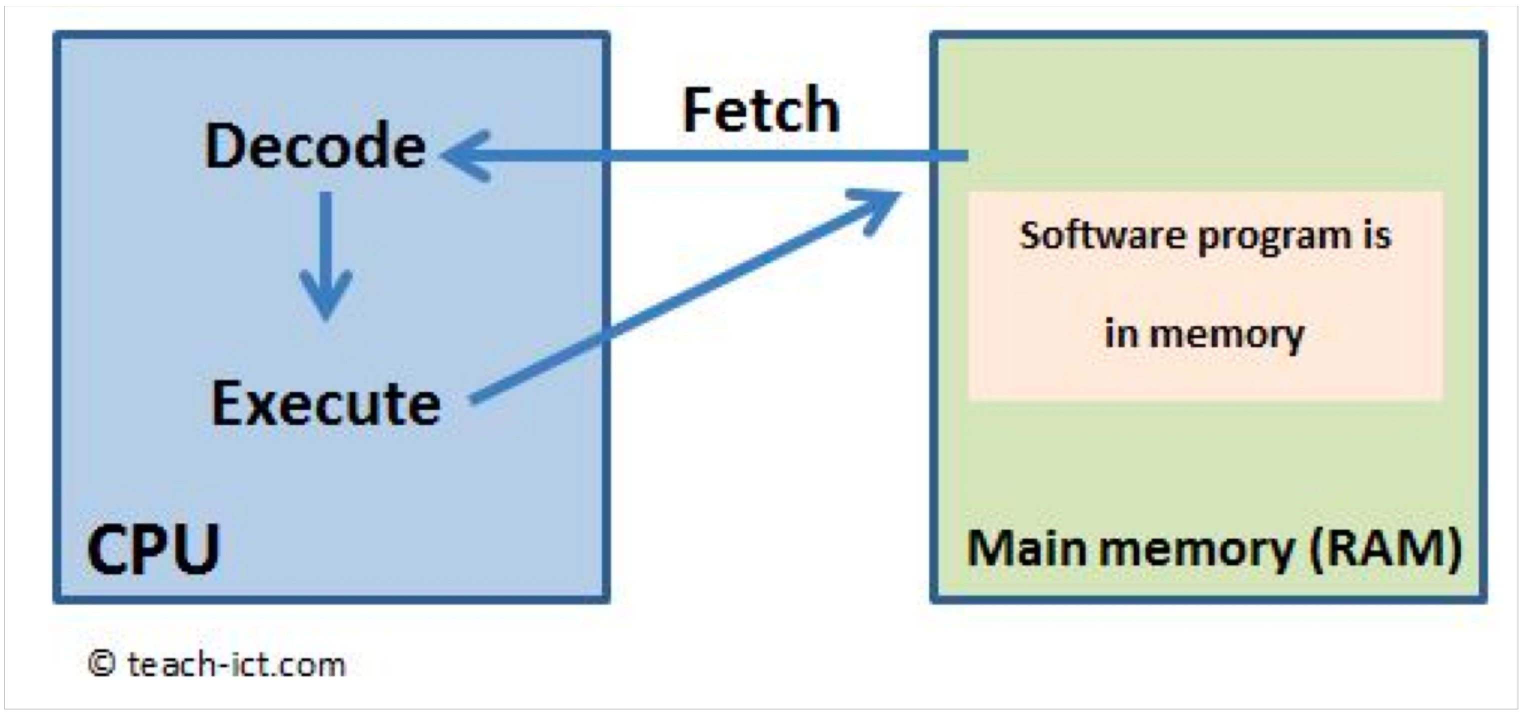
Memory Hierarchy Concepts

- All data are stored at the lowest level
- Data is copied only between *two adjacent levels* at a time



Slide Credit: Dr. Sudhakar Ganti

Fetch Decode Execute Cycle



http://www.teach-ict.com/2016/GCSE_Computing/OCR_J276/1_1_systems_architecture/fetch_execute_cycle/miniweb/index.php

Memory: big bag of bytes…

Memory

Addr 4 bytes	Value 1 byte
0x1000	
0x1001	
0x1002	
0x1003	
0x1004	
0x1005	
0x1006	
0x1007	
0x1008	
0x1009	
0x100a	
0x100b	
0x100c	
0x100d	
0x100e	
0x100f	

Addr 4 bytes	Value 1 byte	Addr 4 bytes	Value 1 byte	Addr 4 bytes	Value 1 byte
0x2000		0x2010		0x2020	
0x2001		0x2011		0x2021	
0x2002		0x2012		0x2022	
0x2003		0x2013		0x2023	
0x2004		0x2014		0x2024	
0x2005		0x2015		0x2025	
0x2006		0x2016		0x2026	
0x2007		0x2017		0x2027	
0x2008		0x2018		0x2028	
0x2009		0x2019		0x2029	
0x200a		0x201a		0x202a	
0x200b		0x201b		0x202b	
0x200c		0x201c		0x202c	
0x200d		0x201d		0x202d	
0x200e		0x201e		0x202e	
0x200f		0x201f		0x202f	

What is stored in memory?

Hint:
Part of design of the Analytical engine!

Types…

- Many things are too big to fit in a single byte
- What is the biggest unsigned integer we could have if we were restricted to a byte?
- What is the range of signed integers we could have if we were restricted to a byte?

Integer data types by size…

# bytes	# bits	Data Type
1	8	char
2	16	short
4	32	int
8	64	long

- What data type do you think an address is?
  - ?
- How many bits do you think an address should be in order to give your CPU enough memory to work with?
  - How many addresses can we make with only 4 bits and therefore how many bytes of memory would we be able to address with 4 bit addresses?
    - ?
  - How many addresses can we make with 8 bits and therefore how many bytes of memory would we be able to address with 1 byte addresses?
    - ?
  - How many addresses can we make with 16 bits and therefore how many bytes of memory would we be able to address with 2 byte addresses?
    - ?
  - How many addresses can we make with 32 bits and therefore how many bytes of memory would we be able to address with 4 byte addresses?
    - ?
  - How many bit addresses would we need to get 8GB of memory addresses?
    - ?

- What data type do you think an address is?
  - Unsigned integer
- How many bits do you think an address should be in order to give your CPU enough memory to work with?
  - How many addresses can we make with only 4 bits and therefore how many bytes of memory would we be able to address with 4 bit addresses?
    - 2^4 = 16 bytes (range 0 to 2^4-1)
  - How many addresses can we make with 8 bits and therefore how many bytes of memory would we be able to address with 1 byte addresses?
    - 2^8 = 256 bytes
  - How many addresses can we make with 16 bits and therefore how many bytes of memory would we be able to address with 2 byte addresses?
    - 2^{16} = 65,536 bytes = 64 KB (1KB = 1024bytes) (64KB = 1024*64 bytes)
  - How many addresses can we make with 32 bits and therefore how many bytes of memory would we be able to address with 4 byte addresses?
    - 2^{32} = 4 GB (1GB = 1024MB) (1MB = 1024KB) (1KB = 1024bytes)
  - How many bit addresses would we need to get 8GB of memory addresses?
    - 2^{33}

Memory: big bag of bytes…

Memory

Addr 4 bytes	Value 1 byte
0x1000	
0x1001	
0x1002	
0x1003	
0x1004	
0x1005	
0x1006	
0x1007	
0x1008	
0x1009	
0x100a	
0x100b	
0x100c	
0x100d	
0x100e	
0x100f	

Addr 4 bytes	Value 1 byte	Addr 4 bytes	Value 1 byte	Addr 4 bytes	Value 1 byte
0x2000		0x2010		0x2020	
0x2001		0x2011		0x2021	
0x2002		0x2012		0x2022	
0x2003		0x2013		0x2023	
0x2004		0x2014		0x2024	
0x2005		0x2015		0x2025	
0x2006		0x201			
0x2007		0x201			
0x2008		0x201			
0x2009		0x201			
0x200a		0x201			
0x200b		0x201			
0x200c		0x201			
0x200d		0x201			
0x200e		0x201			
0x200f		0x201			

Integer data types by size…

# bytes	# bits	Data Type
1	8	char
2	16	short
4	32	int
8	64	long

An architectural decision…

Addr	Value
i	0x1a
i+1	0xb1
i+2	0x45
i+3	0x9e

- Consider 4-byte int
  - it is stored at addresses i, i+1, i+2, and i+3
  - That is, it is at address i and it is 4 bytes long
- Big or Little Endian
  - we start addressing at the BIG END of the number

address	i	i+1	i+2	i+3
Bit positions	2^{31} to 2^{24}	2^{23} to 2^{16}	2^{15} to 2^{8}	2^{7} to 2^{0}

  - or we start at the LITTLE END (Intel)

address	i+3	i+2	i+1	i
Bit positions	2^{7} to 2^{0}	2^{15} to 2^{8}	2^{23} to 2^{16}	2^{31} to 2^{24}