Quiz #1 30/30
HW#1 49.5/50

88%	1st test	25%
	2nd test	25%
	Final	35%
	HW/class participation	15%

ECE 314

- **Microprocessors**
  - programming language (converted)
    - → to assembly language = binary code (no "real" conversion)

0x54
- 0x = base 16, so 54_{16} not 54_{10}

- **Pixels**
  - white ⇒ RGB = 255/255/255 $(11111111_2) = (FF)_{16}$

ex.) 512 kb in base 2
$1k = 2^{10}$
$512 = 2^9$ $= 2^{19}B$

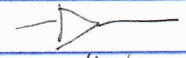
ex.) what's 2^{24}
$2^{20} = 1$ mil ⇒ 16 million
$2^4 = 16$

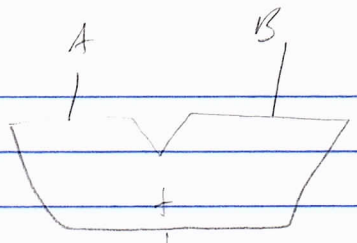
- **Number System**
  - Signed $= [-(2^{n-1}-1), 2^{n-1}-1]$ Twos Comp $= [-2^{n-1}, 2^{n-1}-1]$
  - Unsigned $[0, 2^n-1]$

- **AND/OR vs. NAND/NOR**
  - NAND/NOR req[illegible] hardware to make vs. AND/OR

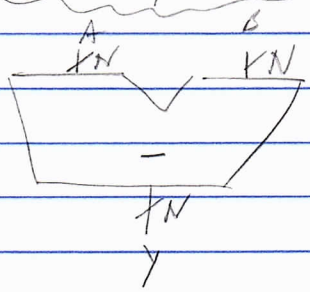
- **Analog vs. [illegible]**
  - Analog
    - [illegible] a multiple ranges
    - 8, 7, 6, 5, 4, 3, 2, 1, 0
    - 0V
    - accepts less noise / small ranges
    - ex.) pin cap throwing
  - Digital
    - 5V, 3V, 2V, 0V
    - '1', '0'
    - accepts lots of noise / large ranges
    - "no such thing as a "true" digital"!

- Slide 68, Ch. 1
  - noise margins - NM

- **Buffer** 
  - "recovers" the voltage

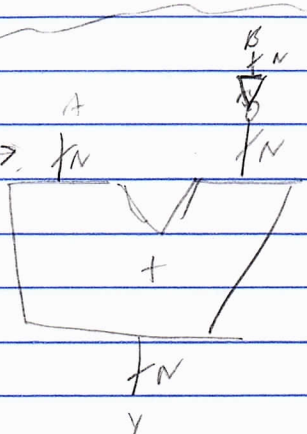


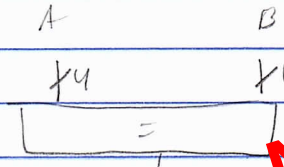
⇐ symbol for adders



∈ symbol for subtractor

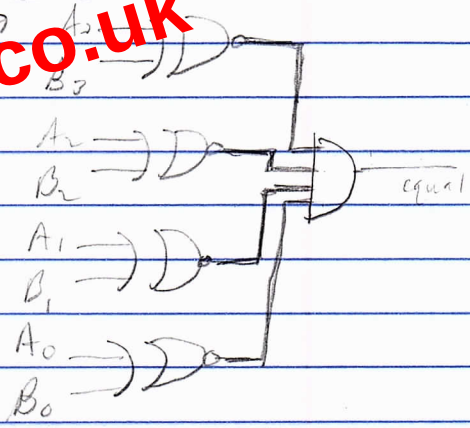
implementation →



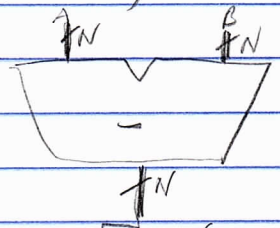


∈ symbol for equality

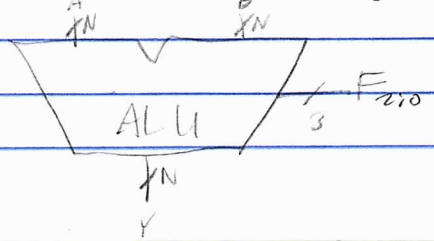
implementation →



if $A < B$, $A - B$ = negative so symbol for less than



ALU (Arithmetic Logic Unit)



F2:0 linked to commands ★

000 = A and B

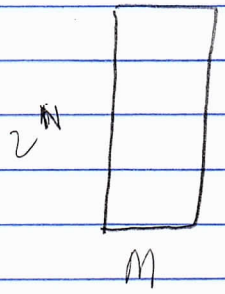
001 = A or B

010 = A + B

etc.

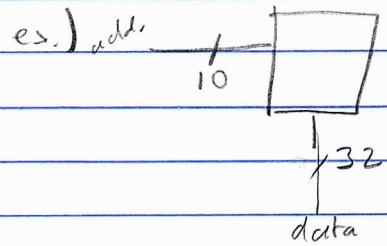
(Ch.5, slide 25)

B = Byte
b = bit



N = # of address bits
M = # of data bits
array size = $2^N \cdot M$

ex.) 3 address bits holding max of 11 bits wide
so N = 3, M = 11
array size = $2^3 \cdot 11 = 88$



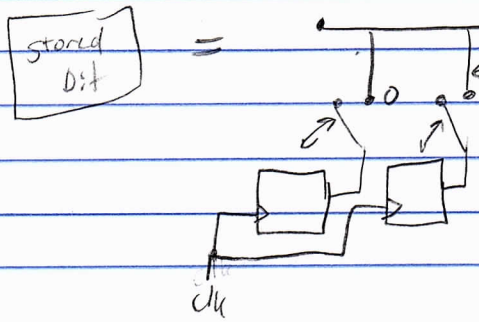
$\Rightarrow$ $2^{10} \cdot 32 = 32K$ bits
$= 2^{15}$ b
$= 2^{12}$ B ($2^{15}/2^3$)
= 4K B

ex.) 8 bit data bus
32 bit address buss
Capacity in bytes $\Rightarrow$ rows = 2^{32} rows
columns = 8 bits
so array size = 32 G bits or 4 GB

Thus, 32-bit systems can only use up to 4GB of memory

Bit Cells for Memory Array

z = bitline due to capacitance charge



wordline = 1 allows data through
wordline = 0 doesn't