I. Fundamentals

- Bit Inversion: XOR with 1 / NOR with itself
- 1's and 2's complement (for negative numbers)

$-4 = -(0100)_2 \Rightarrow (1011)_{1s}$ $(\underline{1}100)_{2s}$

Tips: LSB → MSB [inverse after finding 1]

$\Rightarrow \underbrace{-(2^{n-1}-1)}_{-2^{n-1} \text{ for 2's comp.}}$ to $2^{n-1}-1$

$4 = 4+127 = 131$

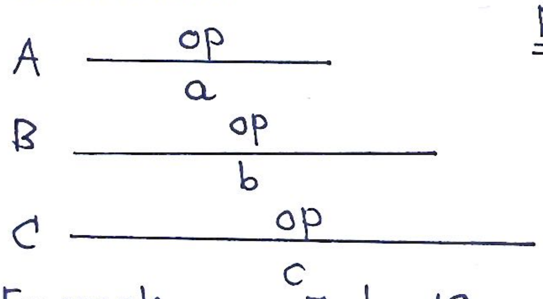
$(\underline{1}0000\overbrace{100}^{\text{add from 127}})$

- Excess Number e.x. excess−127 (IEEE 754)

II. C Programming: struct is passed by value! [copied]

III. ISA: Max and Min Instructions

Variation 1 (No Restriction); $a < b < c$



Max: $\underbrace{2^c - 2^{c-b} - 2^{c-a}}_{C} + \underbrace{1}_{B} + \underbrace{1}_{A}$

Min: Give $2^a - 1$ for A

$[\underbrace{00\ldots0}_{a}$ to $\underbrace{11\ldots10}_{a}]$

$\underbrace{1111\ldots1}_{a}\underbrace{00\ldots0}_{b-a}$ to $\underbrace{11\ldots10}_{b-a} \Rightarrow 2^{b-a}-1$ for B

then 2^{c-b} for C

Example: $a = 7$, $b = 10$, $c = 13$

Max = 8122

Min = 142

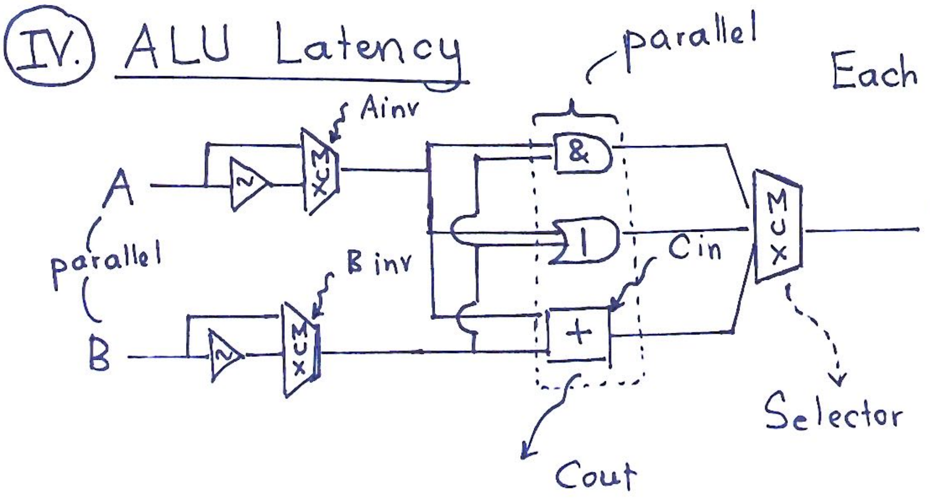
Variation 2 (Give minimum to some format)

Step 1: Determine the minimum for the instructions

Step 2: Allocate the according bits

Step 3: Reduce the problem to variation 1 using the remaining.

IV. ALU Latency



Each slice processes 1 bit at a time

↓

Multiple slices run in parallel

↓

Only wait for the cin from previous bits which causes delay!

(V.) Memorize These! → MIPS things

- No Operation : Avoid register, memory writing and reading.
- Register file = 32 registers (collection) excluding immediate values
- Instruction Register (IR) contains the encoded instructions.
- $sp = last occupied location on top of stack (grows ↓ address)
- PC is updated on the rising edge of the next clock cycle.
- MemToReg multiplexer is reversed because wires cross on diagram!
- **Branch Target Address** (BTA) = $\boxed{PC + 4 + (offset \times 4)}$

(VI.) Control Signal Outputs

Instruction	RegDst	ALUsrc	MemToReg	RegWrite
R-type	1	0	0	1
lw	0	1	1	1
sw	X	1	X	0
beq	X	0	X	0

Instruction	MemRead	MemWrite	Branch	ALUop1	ALUop 2
R-type	0	0	0	1	0
lw	1	0	0	0	0
sw	0	1	0	0	0
beq	0	0	1	0	1