ECE154A — Discussion 01

George Higgins Hutchinson

October 1, 2021

Keep your eyes open for...

- Lab 1: Due Monday, Oct 4
  - Preferred submission format: zip archive with report and code/ancilliary files
  - Alternative submission format: single document with clickable links to code/ancilliary filecontents
- PSet 1: Due Wednesday, Oct 6
- TA Office Hours now set — see website

Parallel Prefix Adder: Why?

- Core observation: anything that is associative can be re-ordered to run in parallel

$$(((A + B) + C) + D) = ((A + B) + (C + D)) \tag{1}$$

- LHS has critical path through 3 gates, RHS through 2
- Remember DeMorgan's Law to handle NOTs

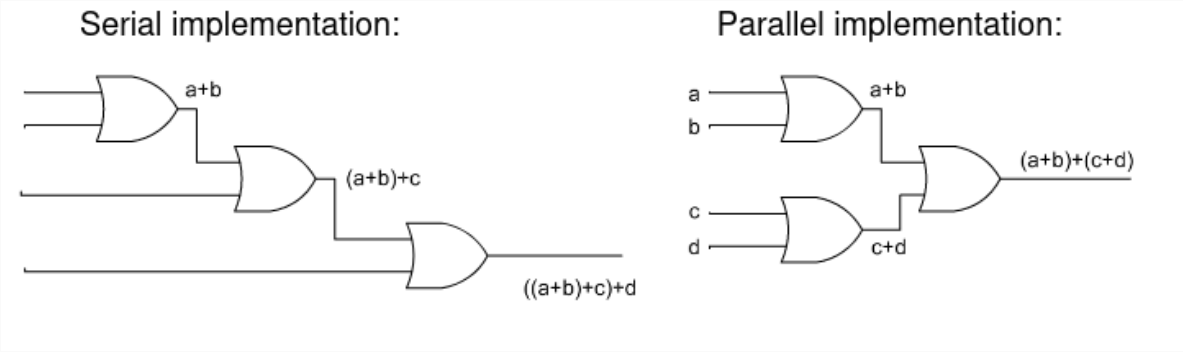


Figure 1: Image credit: Konstas Vitouroulis

Parallel Prefix Adder: Why?

- Given propogate, generate signals, aggregating them is associative
- $(P_{ij}, G_{ij}) = (P_{ik}, G_{ik}) \cdot (P_{kj}, G_{kj})$ for *any* k
- Need $G_{i,-1}$ (does this specific slot have a carry-in for any reason?)
- Pick a reduction that meets your (time, space, power) needs

$$P_{ij} = P_{ik} \wedge P_{kj}$$
$$G_{ij} = G_{ik} \vee (P_{kj} \wedge G_{kj})$$

Parallel Prefix Adder

Ladner-Fischer architecture — delay optimal

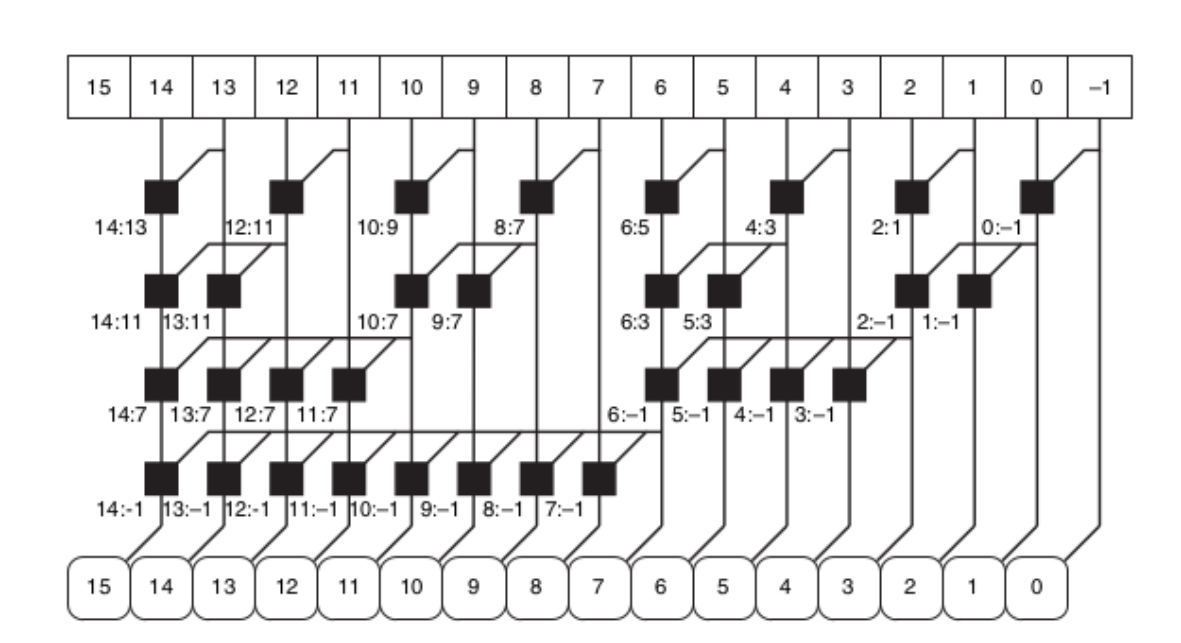


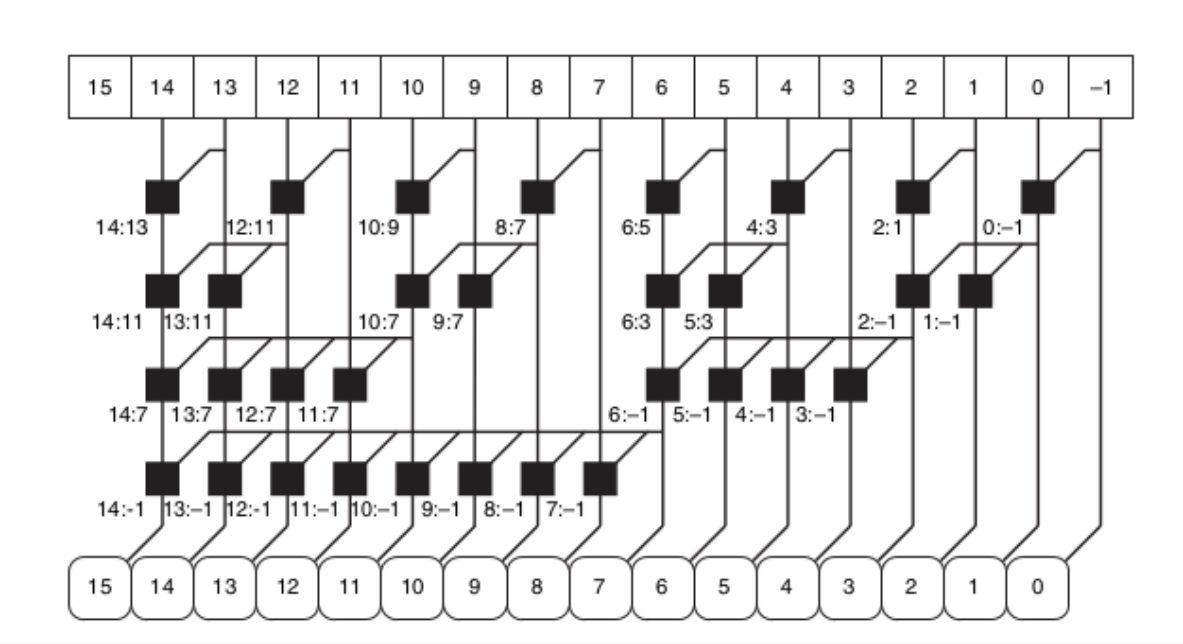
Figure 2: Credit: H&H

What is (are?) the critical path(s)?

Is there some hardware in this figure that could be removed?

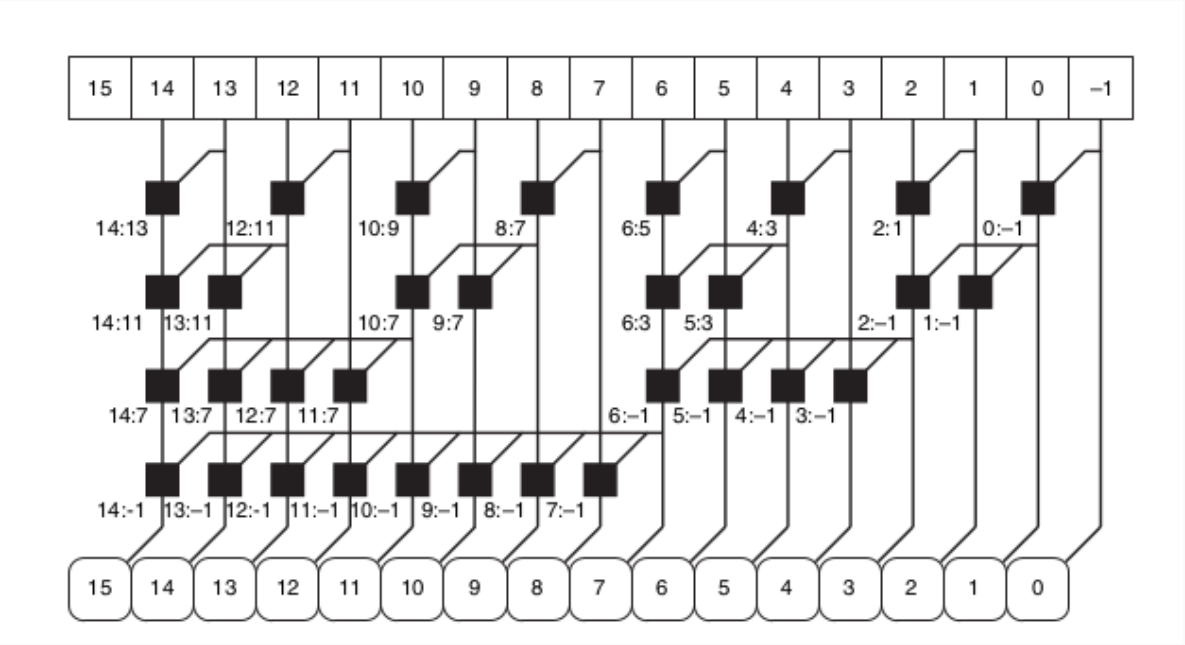
(think about what we're actually trying to compute!)

Parallel Prefix Adder — Critical Path



Critical path: any path through 4 associative blocks

Parallel Prefix Adder — Extra Hardware?



Yes! The adders only care about $G_{i,-1}$, so why make P signals at those nodes?

IEEE-754 Floating Point

	Sign	Exponent	Mantissa	bias
single-precision	1[31]	8[30-23]	23[22-00]	127
double-precision	1[63]	11[62-52]	52[51-00]	1023

$$x = (-1)^{\texttt{sign}} \cdot 2^{\texttt{exp}-\texttt{bias}} \cdot 1.\texttt{mantissa} \tag{2}$$

or, if denorm ($\texttt{exp} = 0$)

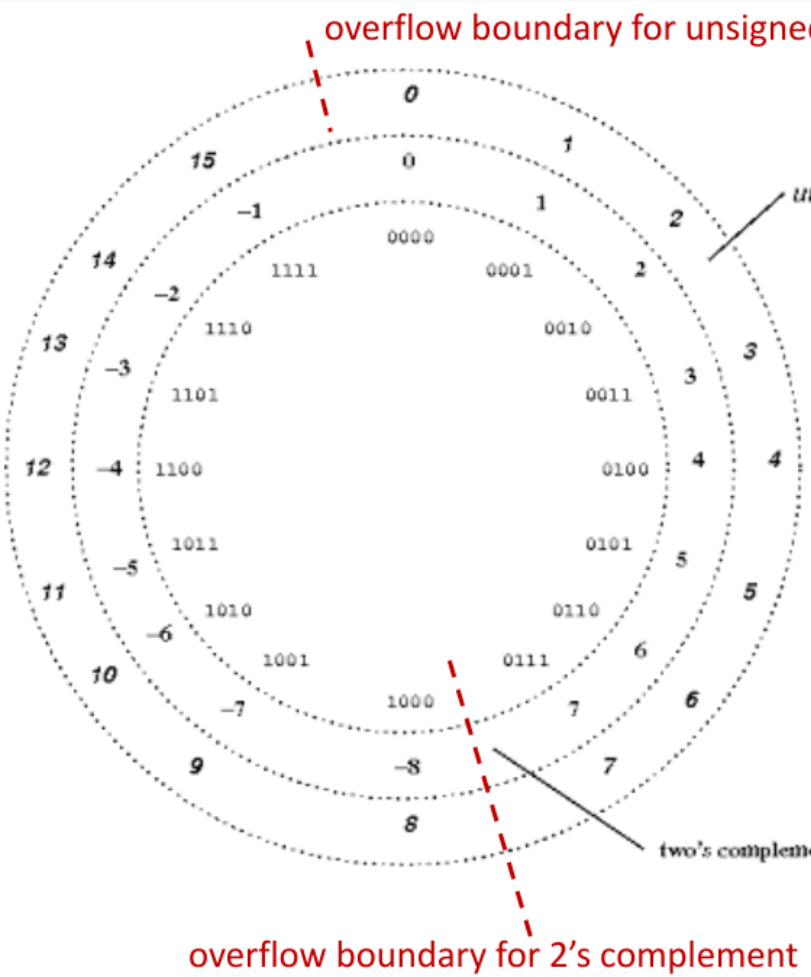
$$x = (-1)^{\texttt{sign}} \cdot 2^{-\texttt{bias}+1} \cdot 0.\texttt{mantissa} \tag{3}$$

$\texttt{exp} = 2 * \texttt{bias} + 1$ for special cases (NaN, inf)

Overflow, Underflow

- Think about your ALU (operates between 32-bit wires)
- What happens to the last carry-out?

Overflow



Cannot fit the results in the given number of bits · overflow

Addition and subtraction are modular operations

Overflow condition depends on the assumed number format and type of operation

Overflow examples

Given 4-bit, unsigned representation add $A = 8$, $B = 8$.

$$\begin{array}{cccccc} & 1 & 0 & 0 & 0 & 0 \\ & & 1 & 0 & 0 & 0 \\ + & 1 & 0 & 0 & 0 & \\ \hline & & 0 & 0 & 0 & 0 \end{array}$$

Given 4-bit, signed representation subtract $A = 2$, $B = -8$.

$$\begin{array}{cccccc} & 0 & 1 & 1 & 1 & 1 \\ & & 0 & 0 & 1 & 0 \\ + & 0 & 1 & 1 & 1 & \\ \hline & & 1 & 0 & 1 & 0 \end{array}$$

Note: this would be right if read as unsigned!

Think about why...