BROWN
School of Engineering

DIGITAL ELECTRONICS SYSTEM DESIGN

FALL 2019

PROF. IRIS BAHAR

NOVEMBER 20, 2019

LECTURE 21: MULTIPLICATION,

ADJUSTED LAB HOURS

- Jarod moved his hours from this Monday to TODAY from 6:30-8:30pm.
  - You have an extra 2 hours to complete your group 2 labs
- Pratistha is canceling her hours this Friday and instead holding additional hours 12-1pm on Tuesday Nov. 26, Dec. 3
- I've updated the calendar

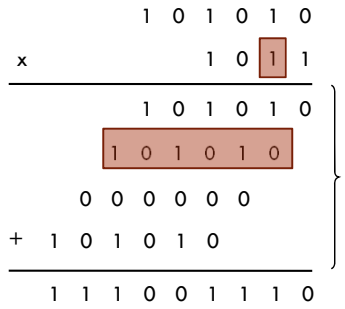
LABS A, B, & C

- Prof. Patterson has set up an FPGA board with a speaker on one of the computers on the south side of the lab.
  - Use this setup for lab B
  - Should work for lab C but still needs to be validated
- Other lab setups will be put out later this week or early next week.

INTEGER MULTIPLICATION REVISITED

- Right shift and add
  - Partial product rows accumulated from top to bottom on an N-bit adder
  - Time for N bits $T_{serial\_mult} = O(N\ t_{adder}) = O(N^2)$ for a RCA
- Making it faster
  - Use a faster adder
  - Use carry-save-adders and avoid carry propagate at each cycle
  - Use multiple adders (array multiplier) with carry save adder cells.
    - Can be easily and efficiently pipelined

THE BINARY MULTIPLICATION



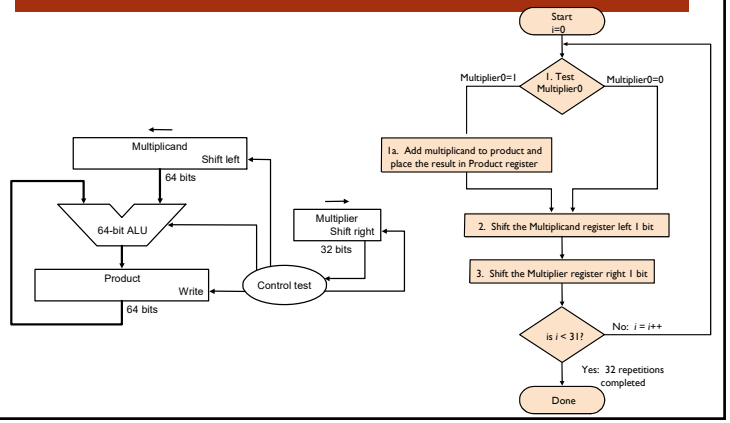
Multiplicand

Multiplier

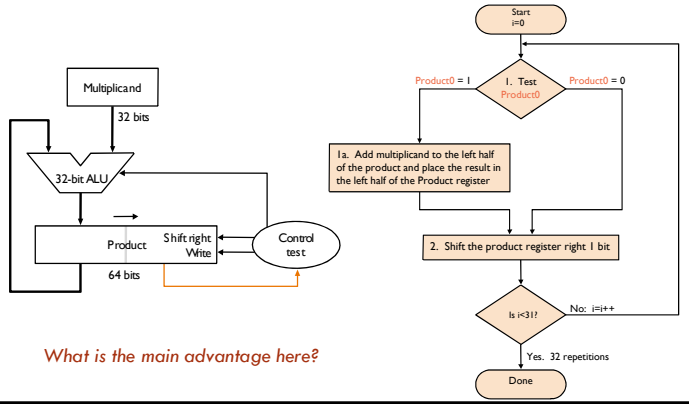
Partial products
(can be formed in parallel)

Double Precision Result

UNSIGNED MULTIPLICATION: VERSION I



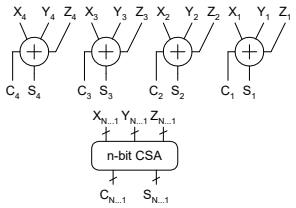
UNSIGNED MULTIPLICATION, VERSION II



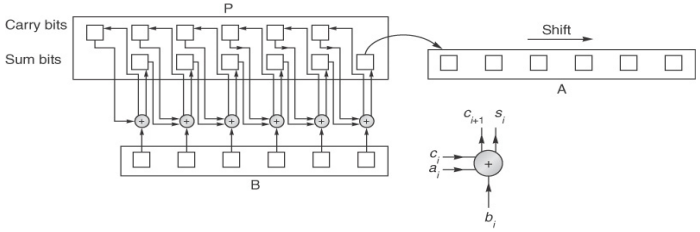
What is the main advantage here?

CARRY SAVE ADDITION

- A full adder in the i^{th} position sums 3 inputs and produces 2 outputs
  - Carry output belongs to $(i+1)^{th}$ bit, sum belongs to i^{th} bit
- N full adders in parallel are called *carry save adder (CSA)*
  - Produce N sums and N carry outs

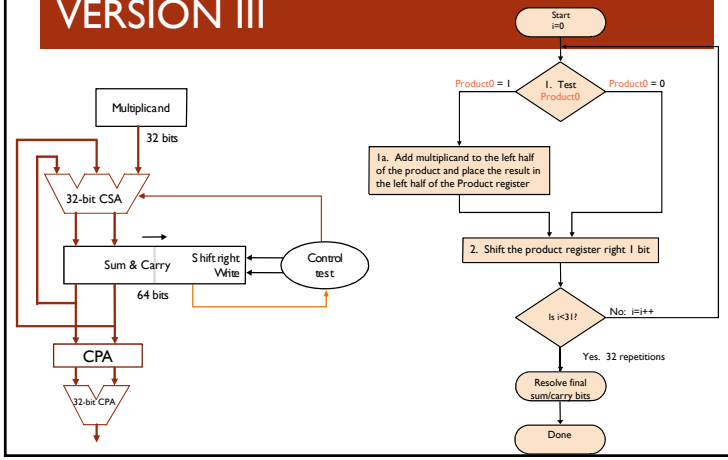


CARRY-SAVE MULTIPLIER

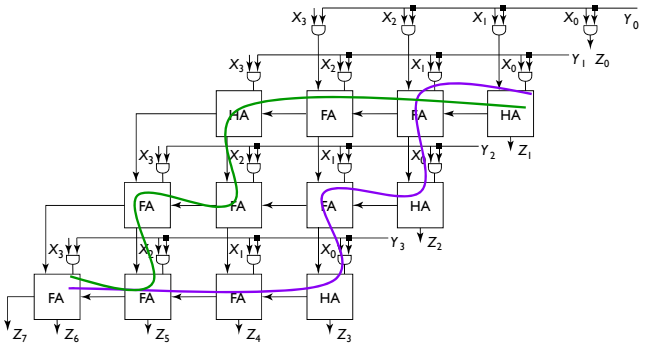


- A single carry-save adder is a collection of *n* independent adders
- Each addition results in a pair of bit vectors, C, S, stored separately in P
- The sum, carry bits of P are fed into the CSA in the following stage, B inputs of CSA are set to 0 if LSB of A=0.
- Requires a separate carry-propagate add at the end to combine the last carry, sum parts
- Still takes *n* cycles to compute, but each stage is faster.
  - Avoid waiting for carry to ripple through at each stage (save for later)

UNSIGNED MULTIPLICATION, VERSION III

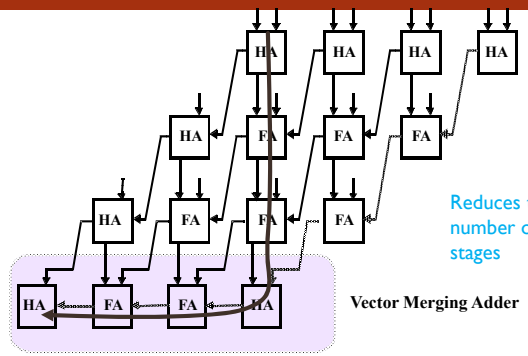


THE ARRAY MULTIPLIER



Finding the critical path is not straightforward !

CARRY-SAVE MULTIPLIER



$t_{mult} = (N-1)t_{carry} + t_{and} + t_{merge}$

PIPELINED ARRAY MULTIPLIER

Can be designed with CSAs or CPAs

Pipelining increases throughput