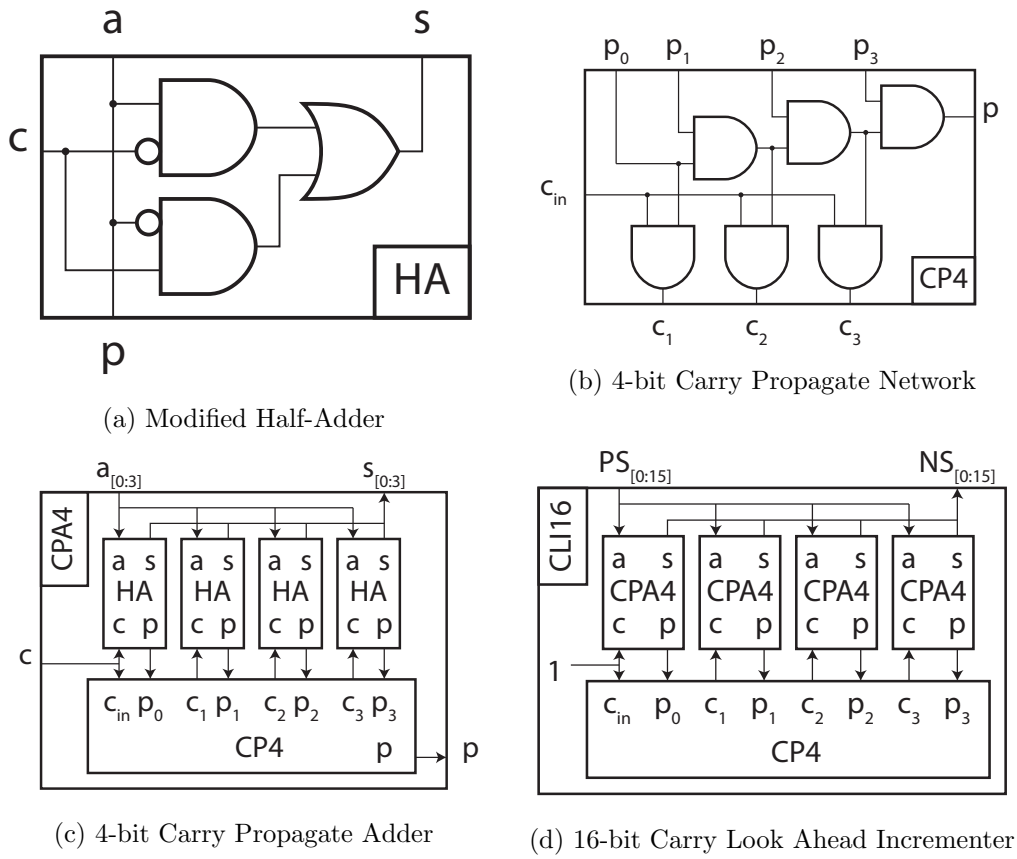
1. **Fast up counter.**

   An up counter has next state decoder NS = PS + 1 . Design a 16 bit"Carry Look Ahead incrementer (add 1) using 4 bit blocks, 2 input gates only. Estimate number of 2 input gates used (assume AND, NAND, OR, NOR, and that inverters can be considered part of 2 input gates). Also estimate worst case delay. Compare to delay and gate count for ripple carry adder and CLA from PS9.1.



(a) Modified Half-Adder

(b) 4-bit Carry Propagate Network

(c) 4-bit Carry Propagate Adder

(d) 16-bit Carry Look Ahead Incrementer

Figure 1: Carry Look Ahead Incrementer Design

Fig. **??** shows the hierarchy of components that make a Carry Lookahead Incrementer. This is fundamentally a carry lookahead adder that has the b-input tied to 0, and the carry in tied to 1. Because b is always 0, there are no generate signals. Buses are split in CPA4 and CLI4 so that the left-most component is tied to the least significant bit of the bus.

The total gate count without simplifying due to the constant 1 c_{in} is

$$GC_{total} = GC_{CP4} + 4\,GC_{CPA4} = GC_{CP4} + 4\,(GC_{CP4} + 4\,GC_{HA}) = 78 \tag{1}$$

With simplification, the lowest HA becomes a NOT gate, the AND gates of the two CP4 networks connected to 1 become wires, and the final AND gates of the p chain in CP4 and CPA4 are not needed, resulting in a total of 69 gates.

The critical path for both cases non-optimized and optimized units is:

Path	Delay	Opt. Delay
$PS_0 \rightarrow CPA4.p$	3	3
$CPA4.p \rightarrow CP4.c_3$	3	2
$CP4.c_3 \rightarrow CPA4.s_3$	3	3
Total	9	8

Compared to the PS9.1 numbers (CLA: 278 gates, 13 delay, Ripple: 144 gates, 34 delay), this incrementer is smaller and faster than both.

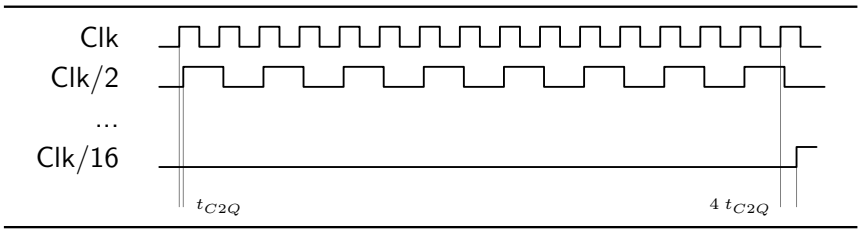
2. **Clock Generation.**

A digital designer group wants to run a FIR filter block at 1/16 the clock speed of the main system clock. The FIR filter block is proposed to connect directly to other blocks in the system without using an asynchronous FIFO. The main system clock, and divided clock are distributed using built in Xilinx global clock buffers.

Note: These answers assume that appropriate valid signal translation logic exists at the clock domain boundaries to prevent duplication or missed data.

(a) Consider clock generation using divide by 16 with a ripple carry counter (Lec 25, slide 28) . Explain with the aid of a timing diagram, issues which might arise either on input or output of FIR module (either data or control conections).

This approach will add a skew of at least 4 t_{C2Q} to the FIR block. This could cause hold violations at the block input, or setup violations at the block output. The following diagram shows how clock skew accumulates at each ripple stage.



(b) Consider clock generation using divide by 16 with a synchronous counter. Explain with the aid of a timing diagram, issues which might arise either on input or output of FIR module.

By using the MSB of a 4-bit counter, this approach would only add a single t_{C2Q} delay to the design, but could still incur the violations mentioned in (a).

(c) Consider clock generation using the builtin Xilinx DCM_ADV clock manager (Chap. 2 of Virtex- 5 FPGA User Guide). How does using this primitive avoid problems seen in a) or b)?

The DCM_ADV manager uses a much higher precision internal clock and feedback network to minimize skew between the input and output clock networks. In general this is the most appropriate way to produce derived clocks.

3. **Soft Errors.**

(Referring to lec26, slide 6, and Xilinx ug116.pdf.) Assume a Virtex-5 design uses 1 Mb for configuration memory and 1 Mb of block RAM. Assume cross-section is in sq.cm.

(a) Assuming 15 neutrons/sq.cm//hr (sea level). How many hours could you expect before the first soft failure is seen in config and block RAM?

UG116 p28 reports 165 FIT/Mb and 692 FIT/Mb for config and block RAM respectively. Taking the reciprocal, one would expect an average time of 691 years between config errors, and 165 years for block RAM.

(b) How can you detect or recover from soft errors in Xilinx Virtex5 FPGA?

Virtex5 devices have dedicated logic for calculating Cyclic Redundancy Checks (CRCs) of memory. If these detect an error, the devices also have deditcated Error Correcting Code (ECC) logic that can repair errors of a a few bits. Alternatively, completely redundant memory can be used to back up and restore the corrupted memory (as used in Triple Modular Redundancy (TMR)).

(c) At 12 km, neutron flux could be 10,000 times larger than at sea level. With a fleet of 1000 aircraft at 12 km equipped with Virtex-5 avionics, how many hours would you expect between soft failures anywhere in the fleet?

Multiplying FIT values from a) by 1000*10,000: 36.4 minutes config, and 8.67 minutes block.

4. **Error Correction.**

Design a single bit error correction, double bit error detection Hamming code for 8 bit data. State the positions of the parity bits, and which bits of the codeword each parity bit protects.

From `http://en.wikipedia.org/wiki/Hamming_code`

Bit position		1	2	3	4	5	6	7	8	9	10	11	12
Encoded data bits		p1	p2	d1	p4	d2	d3	d4	p8	d5	d6	d7	d8
Parity bit coverage	p1	X		X		X		X		X		X	
	p2		X	X			X	X			X	X	
	p4				X	X	X	X					X
	p8								X	X	X	X	X

5. **Power/Energy.**

Designer A decides to duplicate her datapath in an accelerator such that the compute throughput can be increased by 2X (you can assume the application contains enough parallelism for this to happen) when she runs the accelerator at the same clock frequency.

(a) Designer B tries to match this performance gain by increasing the voltage of his circuit (assume the max frequency a CMOS circuit is positively related to the voltage at which it runs). In terms of dynamic power consumption, do you think his solution is better or worse than that of Designer A ? Why?

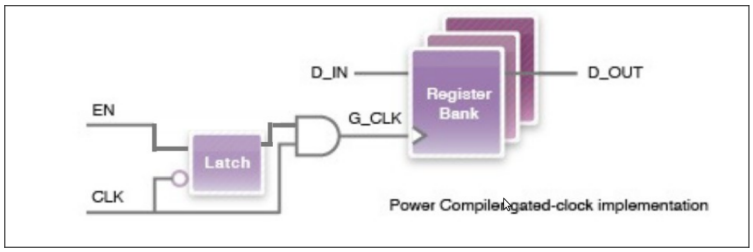
Dynamic power consuption is porpotional to V^2. Designer A will double the dynamic power consuption, while designer B will quadruple it.

(b) With the duplicated datapath, Designer A can afford to reduce the clock frequency by half yet still achieve the throughput of the original accelerator. If she does not lower the voltage of the circuit, how does this reduction of clock frequency effect the overall power of the circuit? How does it effect the overall energy consumption for running the application? Explain your reasoning. (Note: the baseline used in this question is the accelerator with a duplicated datapath.

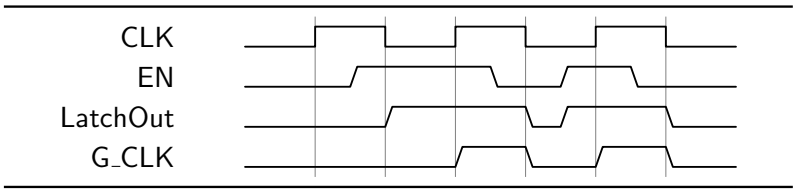
Dynamic consuption is proportional to frequency, so the power consuption is reduced by half. However, running the whole application will take twice as long compared to the original duplicated datapath, so the the application energy consuption is the same.

6. **Power Down circuit.**

Gating the clock is generally a very bad idea. However disabling the clock can drastically reduce power consumption for a block of circuitry. The circuit below claims to be a safe way to generate G_CLK. Show with a timing diagram whether this is true or not.

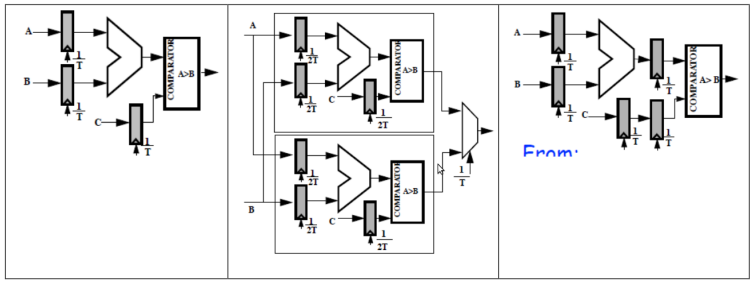


The design shown is standard clock gating circuit for positive edge logic. Below shows its glitch-prevention operation. Note that EN is connected to the data input of the latch.



7. **Architecture for Power reduction.**

A. Chandrakasan (CS150 F89) and R. Brodersen [IEEE 1995] proposed three architectures for a sum/comparator. Assuming delay Tdq, Tsum, Tcomp and Tsu, determine throughput and minimum clock period for each design.



Throughput is normalized against the simple design.

Architecture	Throughput	Min. Period
Simple	1	$T_{dq} + T_{sum} + T_{comp} + T_{su}$
Parallel	2	$T_{dq} + T_{sum} + T_{comp} + T_{mux} + T_{su}$
Pipelined	1	$T_{dq} + min(T_{sum}, T_{comp}) + T_{su}$