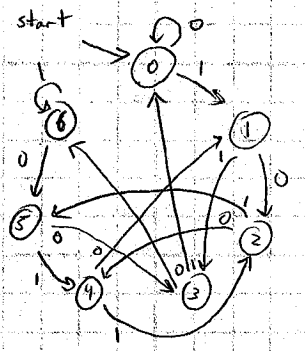
2006 Bits come in serially, 1 at a time LSB first.

a) What is the state diagram to output 1 if the number is divisible by 7?



OUT = 1 in state 0

101
110
100
101
1101

b) Because 7 is a prime number, we believe 7 is the minimum number of states.

c) To be divisible by 28, it must be divisible by both 7 and 4. One option would be to add another state machine of 4 states (to check divisibility by 4) and set **OUT = (divisible by 7) AND (divisible by 4)**

d) One algorithm for determining primality is for an input n, check if any prime integer m from 2 to $\sqrt{n}$ divides evenly. If none, then it is a prime. (See Primality Tests) But, as for us implementing this algorithm in a state machine, we have no solution because there are no bounds on the input.

2007 $S = 0, -1, 1, -2, 4, -8, 16, -32 \ldots$ e.g. 0 1 1 0 = 2 (16 −8 4 −2 1)

1. How to convert to binary without going to decimal?

Given a number $S \rightarrow (S \,\&\, 0x5555\ldots) - (S \,\&\, 0xAAAA\ldots)$

(positive portion − negative portion)

example: $S = 0110$ $\quad S \,\&\, 0x5 = 0100$ $\quad S \,\&\, 0xA = 0010$

$$\begin{array}{r} 0100 \\ -\,0010 \\ \hline 0010 \end{array}$$

2. Binary to S? (e.g. 0110 → 11010)

$S = (\text{binary} + 0xAAAA\ldots) \oplus 0xAAAA\ldots$

Example $\begin{array}{r} 0110 \\ +\,1010 \\ \hline 10000 \end{array}$ $\quad \begin{array}{r} 10000 \\ \oplus\,01010 \\ \hline 11010 \end{array}$

3. To add, every time a carry bit is generated, propagate an extra carry to the next bit.

4. It can take an extra bit to represent a number than binary (e.g. 6 is 4 bits in binary but 5 in S) Harder to optimize multiplication into shifts. May take additional hardware to make an ALU The main issue seems to be the increased complexity of arithmetic ops.

5. Yes, for integers. Non-unique representations exist for rationals.

Assuming positive integers...

"Negative Base" on Wikipedia

MITRA | EE QUALS | LUKE HSIAO | Dec 2015

2009

1. Combinational: Output depend ONLY on the present values of the input
   Sequential: Output depends on both present and past values of inputs.

2. It seems that if we use a strict definition of combinational, it may not be. But, because the wrap around carry will stabilize after a 2nd pass, we can consider it a combinational circuit, so long as it is given enough time to stabalize.

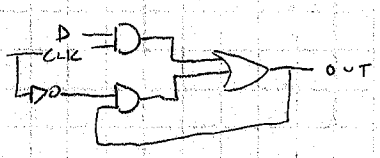
3. It's a stretch, but we could loosen our definition to accepting things as combinational if their outputs become stable in time to not violate any setup/hold times in the rest of the circuit.

4. For this adder, we would have to measure the time from the first bit with an unknown carry in, through the rest of the bits, then back through again with a valid carry bit.

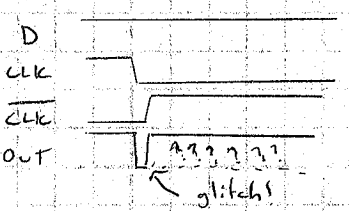
5. In general, we just need to be able to time from when input changes until the output settles and becomes stable. For arbitrary circuits with loops, it would be difficult to determine how long it will take to settle, especially since that time may depend on the inputs.

2010

1. A latch is transparent (value passes through) while enabled, a flip flop is not. A flip-flop samples the input value at a specific time.

2. 

At a high-level, this works correctly. But, it has a hazard on the CLK path. For example, if D=1 and OUT=1



3. To fix this, we can add more logic. Specifically, by adding $D \cdot OUT$

D OUT \ CLK	0	1
0 0	0	0
0 1	1	0
1 1	1	1
1 0	0	1

So that $OUT = D \cdot CLK + \overline{CLK} \cdot OUT + D \cdot OUT$.

4. It seems the idea here is to utilize the gate's delay

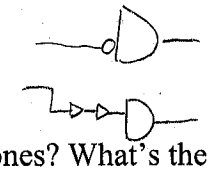
5. ???

Prof. Subhasish Mitra 2013

Name

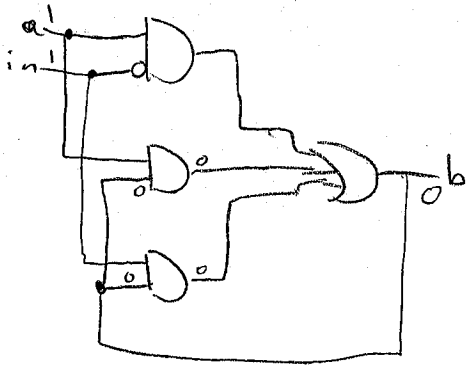
Start time

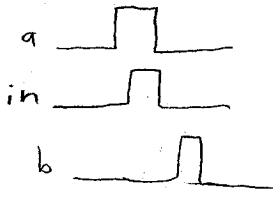
1. What are hazards in a logic circuit?
   Hazards in a circuit may produce glitches. This can be a static glitch, or a glitch that involves an output changing multiple times in response to a single input change. Or a functional hazard if multiple inputs change.
2. Is there a hazard problem for output "a" of the circuit below?
   Yes! Note the reconverging path of in and $\overline{in}$.
   $A = A\overline{in} + in\overline{B}$
3. If yes, please fix it. If not, why not?
   $A = in\overline{B} + \overline{B}A + \overline{in}A$. Alternatively, add delays.



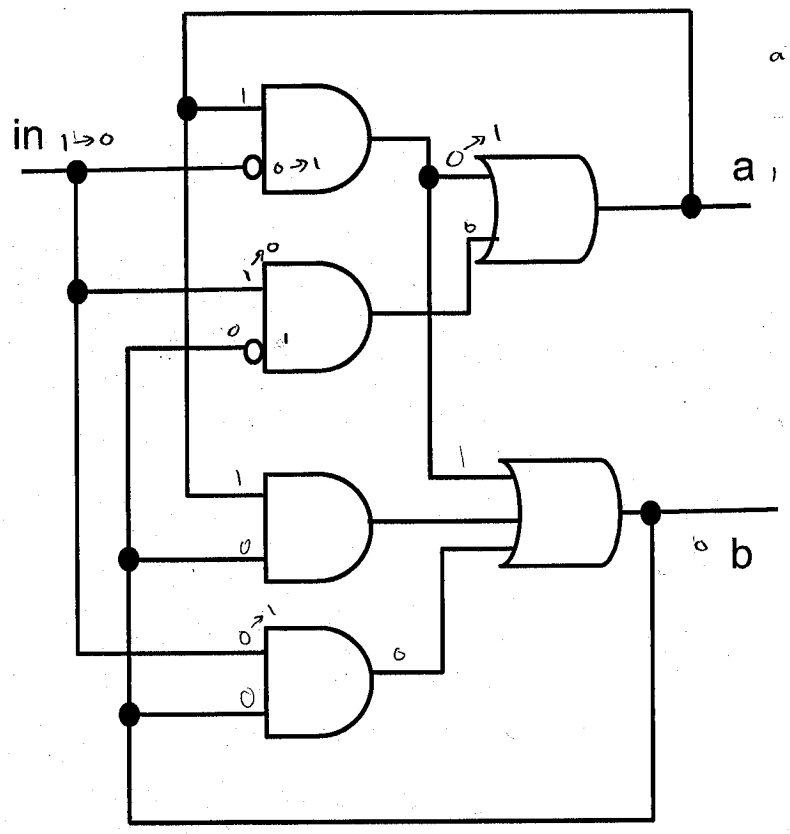
4. Is there any other hazard problem in this circuit? Which ones? What's the fix? Is it possible to fix those problems without inserting delays?
   $B = \overline{in}A + BA + Bin \rightarrow$ No hazard!
   However, can there be functional (2-input change) hazards? Yes!
   In fact, for both A and B! To eliminate need to balance the delays.

* Adding mask gates increases power more than the glitch and makes it harder to test.





Name



State	Code	Next	
	b a	in = 0	in = 1
A	0 0	(A)	B
B	0 1	C	(B)
C	1 1	(C)	D
D	1 0	A	(D)

Truth table for a

b a \ in	0	1
0 0		1
0 1	1	1
1 1	1	
1 0		

Truth table for b

b a \ in	0	1
0 0		
0 1	1	
1 1	1	1
1 0		1