

**Lecture 24 –
 Latches**



Lecturer PSOE Dan Garcia

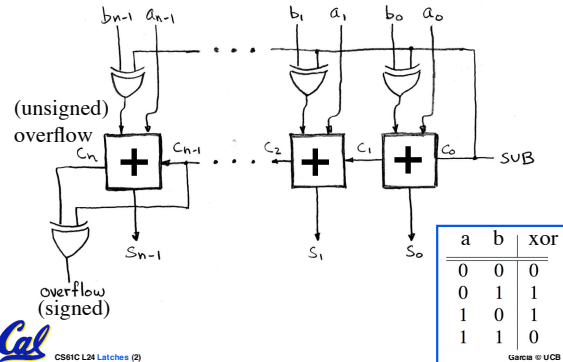
www.cs.berkeley.edu/~ddgarcia

SETI@Home? Yak! =>

The Honeynet project uses "honeypot" PCs & monitors how long it takes (sec-min) for them to be hacked and what happens. 1 Mebi "botnets" are used for spam, viruses, DDoS, Google AdSense, hacking gambling, id theft!

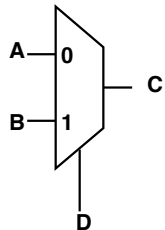


Last time: Extremely Clever Subtractor



2-Input Multiplexor (MUX) Review

Symbol



Definition

D	C
0	A
1	B



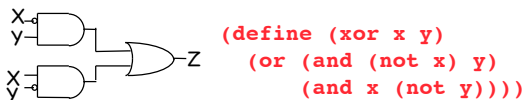
Review...

- Use muxes to select among input
 - S input bits selects 2^S inputs
 - Each input can be n-bits wide, indep of S
- Implement muxes hierarchically
- ALU can be implemented using a mux
 - Coupled with basic block elements
- N-bit adder-subtractor done using N 1-bit adders with XOR gates on input
 - XOR serves as conditional inverter



Combinational Logic from 10 miles up

- CL circuits simply compute a binary function (e.g., from truthtable)
- Once the inputs go away, the outputs go away, nothing is saved, **no STATE**
 - Similar to a function in Scheme with no set! or define to save anything

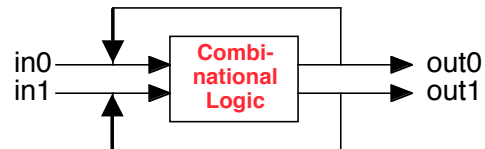


- How does the computer remember data? [e.g., for registers]



State Circuits Overview

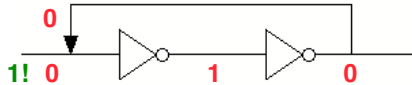
- State circuits have feedback, e.g.



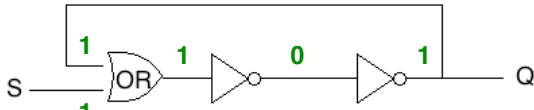
- Output is function of inputs + fed-back signals.
- Feedback signals are the circuit's state.
- What aspects of this circuit might cause complications?



A simpler state circuit: two inverters



- When started up, it's **internally stable**.
- Provide an **OR** gate for coordination:



What's the result? How do we set to 0?



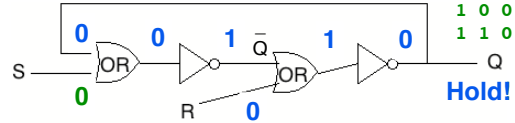
CS61C L24 Latches (7)

Garcia © UCB

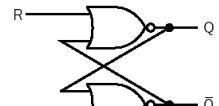
An R-S latch (cross-coupled NOR gates)

- S means "set" (to 1), R means "reset" (to 0).

A	B	NOR
0	0	1
0	1	0
1	0	0
1	1	0



- Adding Q' gives standard RS-latch:



Truth table

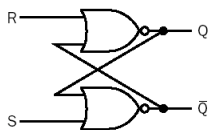
S	R	Q
0	0	hold (keep value)
0	1	0
1	0	1
1	1	unstable



CS61C L24 Latches (8)

Garcia © UCB

An R-S latch (in detail)



Truth table

S	R	Q	Q̄	Q(t+Δt)
0	0	0	1	0 hold
0	0	1	0	1 hold
0	1	0	1	0 reset
0	1	1	0	0 reset
1	0	0	1	1 set
1	0	1	0	1 set
1	1	0	0	x x unstable
1	1	1	1	x x unstable

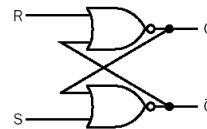
A	B	NOR
0	0	1
0	1	0
1	0	0
1	1	0



CS61C L24 Latches (9)

Garcia © UCB

R-S latch in scheme



It's really just...

recursion!
(demo)

A	B	NOR
0	0	1
0	1	0
1	0	0
1	1	0

```
(define (rs-latch r s)
  (define (rs1-h q qbar)
    (rs1-h (nor r qbar)
           (nor q s)))
  (rs1-h #t #f))
```

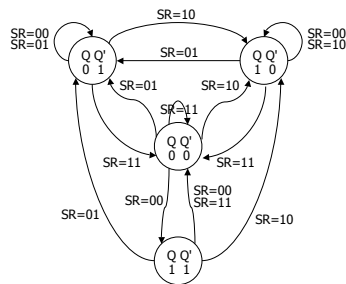


CS61C L24 Latches (10)

Garcia © UCB

State diagram

- States represent possible output values.
- Transitions represent changes between states based on inputs.



CS61C L24 Latches (11)

Garcia © UCB

What does it mean to "clobber" midterm?

- You **STILL** have to take the final even if you aced the midterm!
- The final will contain midterm-material Qs and new, post-midterm Qs
- They will be graded separately
- If you do "better" on the midterm-material, we will **clobber your midterm with the "new" score!** If you do worse, midterm unchanged.
- What does "better" mean?
 - Better w.r.t. Standard Deviations around mean
- What does "new" mean?
 - Score based on remapping St. Dev. score on final midterm-material to midterm score St. Dev.



CS61C L24 Latches (12)

Garcia © UCB

“Clobber the midterm” example

• Midterm

- Mean: 47
- Standard Deviation: 14
- You got a 33, one σ below



• Final Midterm-Material Questions

- Mean: 40
- Standard Deviation: 20
- You got a 60, one σ above
- Your new midterm score is now mean + σ
= 47 + 14 = 61 (~ double your old score)!

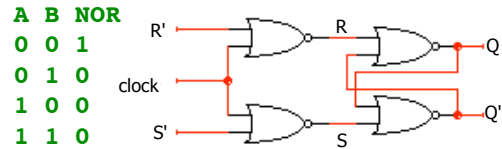


CS61C L24 Latches (13)

Garcia © UCB

Controlling R-S latch with a clock

- Can't change R and S while clock is active.



- Clocked latches are called *flip-flops*.



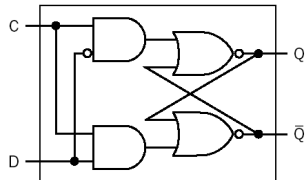
CS61C L24 Latches (14)

Garcia © UCB

D flip-flop are what we really use

- Inputs C (clock) and D.
- When C is 1, latch open, output = D (even if it changes, “transparent latch”)
- When C is 0, latch closed, output = stored value.

C	D	AND
0	0	0
0	1	0
1	0	0
1	1	1

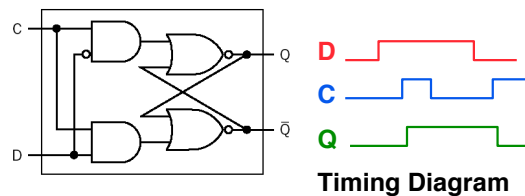


CS61C L24 Latches (15)

Garcia © UCB

D flip-flop details

- We don't like transparent latches
- We can build them so that the latch is only open for an instant, on the **rising edge of a clock** (as it goes from 0→1)



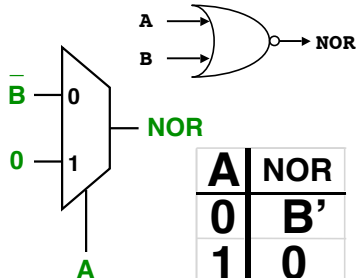
CS61C L24 Latches (16)

Garcia © UCB

But do you really understand NORs?

- If one input is 1, what is a NOR?
- If one input is 0, what is a NOR?

A	B	NOR
0	0	1
0	1	0
1	0	0
1	1	0



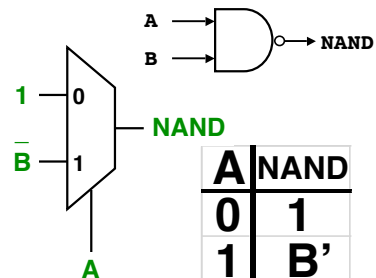
CS61C L24 Latches (17)

Garcia © UCB

But do you really understand NANDs?

- If one input is 1, what is a NAND?
- If one input is 0, what is a NAND?

A	B	NAND
0	0	1
0	1	1
1	0	1
1	1	0



CS61C L24 Latches (18)

Garcia © UCB

Peer instruction

Pick the truth table that results from substituting NAND gates for the NOR gates in the R-S latch:

<table border="0"> <tr><td>A</td><td>B</td><td>NOR</td></tr> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </table>	A	B	NOR	0	0	1	0	1	0	1	0	0	1	1	0		<table border="0"> <tr><td>A</td><td>B</td><td>NAND</td></tr> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </table>	A	B	NAND	0	0	1	0	1	1	1	0	1	1	1	0
A	B	NOR																														
0	0	1																														
0	1	0																														
1	0	0																														
1	1	0																														
A	B	NAND																														
0	0	1																														
0	1	1																														
1	0	1																														
1	1	0																														

<table border="0"> <tr><td>A</td><td>B</td><td>Q</td></tr> <tr><td>0</td><td>0</td><td>hold</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>undef</td></tr> </table>	A	B	Q	0	0	hold	0	1	0	1	0	1	1	1	undef	<table border="0"> <tr><td>A</td><td>B</td><td>Q</td></tr> <tr><td>0</td><td>0</td><td>hold</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>undef</td></tr> </table>	A	B	Q	0	0	hold	0	1	1	1	0	0	1	1	undef	<table border="0"> <tr><td>A</td><td>B</td><td>Q</td></tr> <tr><td>0</td><td>0</td><td>undef</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>hold</td></tr> </table>	A	B	Q	0	0	undef	0	1	0	1	0	1	1	1	hold	<table border="0"> <tr><td>A</td><td>B</td><td>Q</td></tr> <tr><td>0</td><td>0</td><td>undef</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>hold</td></tr> </table>	A	B	Q	0	0	undef	0	1	1	1	0	0	1	1	hold
A	B	Q																																																													
0	0	hold																																																													
0	1	0																																																													
1	0	1																																																													
1	1	undef																																																													
A	B	Q																																																													
0	0	hold																																																													
0	1	1																																																													
1	0	0																																																													
1	1	undef																																																													
A	B	Q																																																													
0	0	undef																																																													
0	1	0																																																													
1	0	1																																																													
1	1	hold																																																													
A	B	Q																																																													
0	0	undef																																																													
0	1	1																																																													
1	0	0																																																													
1	1	hold																																																													

1 2 3 4



CS61C L24 Latches (19)

Garcia © UCB

Peer Instruction

- A. $(a+b) \cdot (\bar{a}+b) = b$
- B. N-input gates can be thought of cascaded 2-input gates. I.e., $(a \Delta bc \Delta d \Delta e) = a \Delta (bc \Delta (d \Delta e))$ where Δ is one of AND, OR, XOR, NAND
- C. You can use NOR(s) with clever wiring to simulate AND, OR, & NOT

ABC
1: FFF
2: FFT
3: FTF
4: FTT
5: TFF
6: TFT
7: TTF
8: TTT



CS61C L24 Latches (20)

Garcia © UCB

Peer Instruction

- A. Truth table for mux with 4-bits of signals has 2^4 rows
- B. We could cascade N 1-bit shifters to make 1 N-bit shifter for sll, srl
- C. If 1-bit adder delay is T, the N-bit adder delay would also be T

ABC
1: FFF
2: FFT
3: FTF
4: FTT
5: TFF
6: TFT
7: TTF
8: TTT



CS61C L24 Latches (24)

Garcia © UCB

“And In conclusion...”

- We use feedback to maintain **state**
- RS-Latch the simplest memory element
 - We're not allowed to assert both R and S
- Clocks tell us when latches change
- D-FlipFlops used to build Register files



CS61C L24 Latches (26)

Garcia © UCB