Digital logic: PA6 quickstart, decoders, multiplexers, sequential logic

Yipeng Huang

Rutgers University

April 27, 2021

Table of contents

Looking ahead

Class plan

1. PA5 due tomorrow, Wednesday, 4/27.
2. PA6 releases this evening, due Tuesday, 5/11. Worth 40 points, with opportunity for 40 points extra credit.
3. As announced at beginning of semester, there is no final exam in this course.

Final course rubric

Quizzes

- 20% of course raw score from the 5 graded quizzes.
- In other words, each graded quiz was worth 4% of course raw score.

Programming assignments

- 80% of course raw score from the 6 programming assignments.
- PA1 through PA5 graded out of 120 points.
- PA1 through PA5 each worth 15% of course raw score.
- PA6 graded out of 40 points with opportunity for 40 point extra credit.
- PA6 worth 5% of course raw score with opportunity for 5% extra credit raw score.

Table of contents

PA6 Quickstart: Input file format

Input output count and name declaration

- INPUTVAR <N> input0 input1 ... input<N-1>
- OUTPUTVAR <M> output0 output1 ... output<M-1>

Single input gate (NOT gate) declaration

- NOT input0 output0

Double input gate (AND, NAND, OR, NOR, XOR, XNOR gates) declaration

- OR input0 input1 temp0
- NAND temp0 temp0 output0

PA6 Quickstart: Output format

Print truth table for input and output variables

- Omit intermediate temp variables.
- For N inputs, print 2^N rows for all combinations of input variable assignments.
- Columns 0 to N-1 correspond to input0, input1, ... input<N-1>
- Columns N to N+M-1 correspond to output0, output1, ... output<M-1>

PA6 Quickstart: Example input output pair

```
input0 input1 | temp0 | output0
---------------------------------
0      0      | 0     | 1
1      0      | 1     | 0
0      1      | 1     | 0
1      1      | 1     | 0
```

I. Suppose:
N inputs
M outputs

Truth table size:
2^N rows
N+M columns

II.
NAND = not AND
!(input0 && input1)

III.
Unknown number of gates and unknown number of temp variables.
Can't use arrays.
So use linked list to track temp variables.
Link list keeps track of pairs of:
(names of vars, and values of variables)

Table of contents

Combinational vs. sequential logic

Combinational logic

- No internal state nor memory
- Output depends entirely on input
- Examples: NOT, AND, NAND, OR, NOR, XOR, XNOR gates, decoders, multiplexers.

Using a hardware design language like Verilog or VHDL, a computer engineer would use these gates to build up arithmetic units like addition, subtraction in a CPU.

Sequential logic

- Has internal state (memory)
- Output depends on the inputs and also internal state
- Examples: latches, flip-flops, Mealy and Moore machines, registers, pipelines, SRAMs.

Decoders

Takes n-bit input, uses it as an index to enable exactly one of 2^n outputs

Internal design of 1:2 decoder

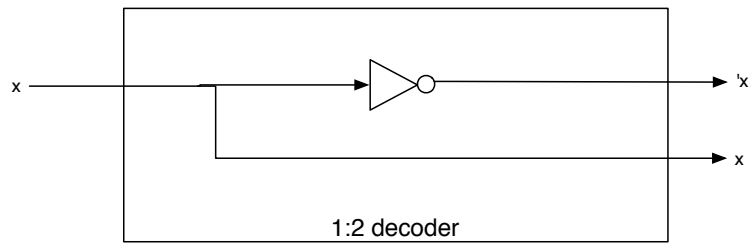


Figure: Source: Mano & Kime

Decoders

Hierarchical design of decoder (2:4 decoder)

Takes n-bit input, uses it as an index to enable exactly one of 2^n outputs

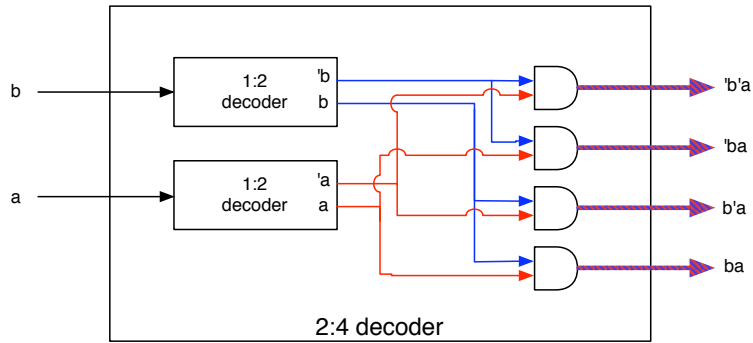


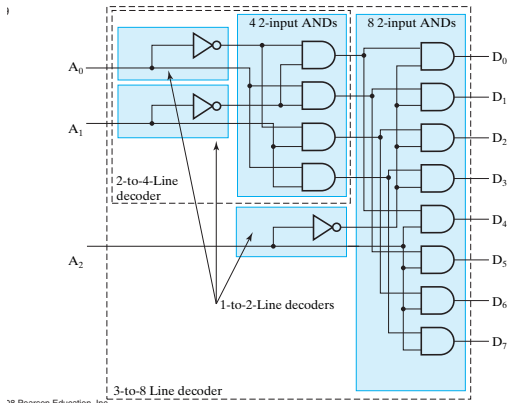
Figure: Source: Mano & Kime

Decoders

Takes n-bit input, uses it as an index to enable exactly one of 2^n outputs

Decoder (3:8)

Hierarchical design: use small decoders to build bigger decoder



Note: A_2 "selects" whether the 2-to-4 line decoder is active in the top half (A_2=0) or the bottom (A_2=1)

Figure: Source: Mano & Kime

Multiplexers

Using n-bit selector input, select among one of 2^n choices

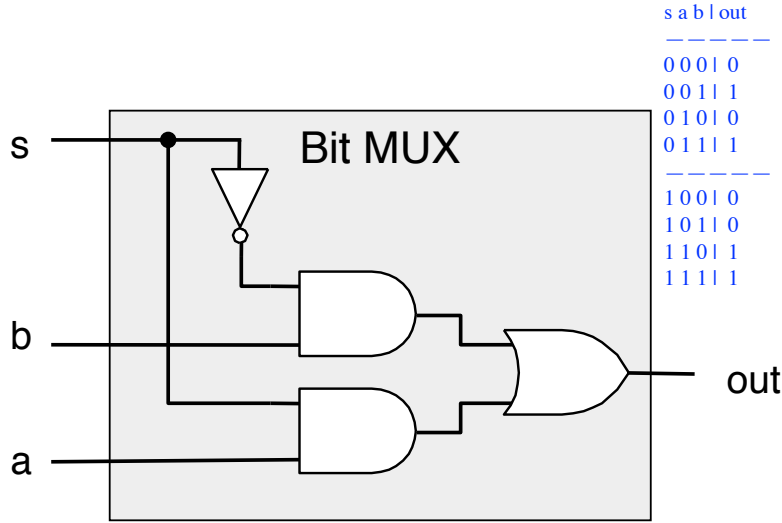


Figure: Source: CS:APP

Multiplexers

Using n-bit selector input, select among one of 2^n choices

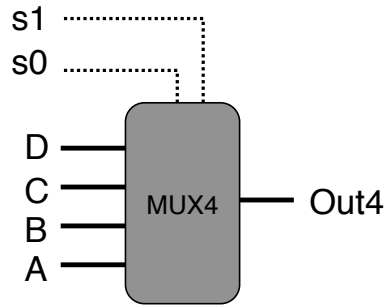


Figure: Source: CS:APP

Multiplexers

Using n-bit selector input, select among one of 2^n choices

SystemVerilog and VHDL, specify these wires, multiplexers, decoders, and gates, run these designs through electronic design automation tools. The EDA tools minimize the logic using e.g., Karnaugh maps to create minimized combinational logic.

Internal mux organization

3-26

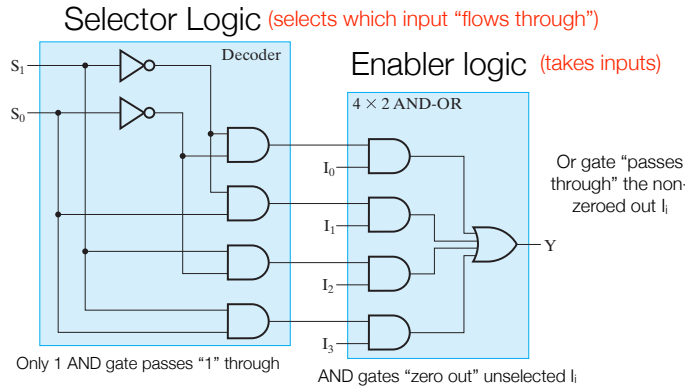


Figure: Source: Mano & Kime

Putting all combinational logic together: Seven-segment display

Table of contents

Sequential logic

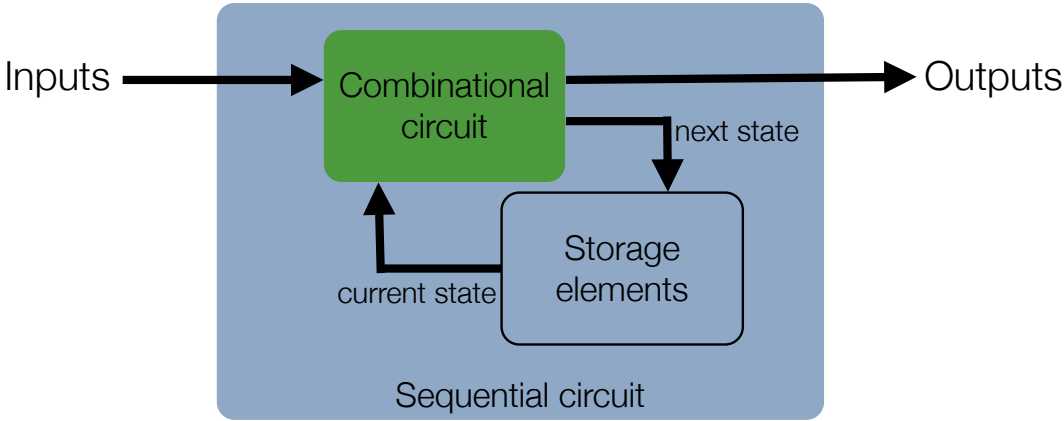


Figure: Source: Mano & Kime

Combinational vs. sequential logic

Combinational logic

- No internal state nor memory
- Output depends entirely on input
- Examples: NOT, AND, NAND, OR, NOR, XOR, XNOR gates, decoders, multiplexers.

Sequential logic

- Has internal state (memory)
- Output depends on the inputs and also internal state
- Examples: latches, flip-flops, Mealy and Moore machines, registers, pipelines, SRAMs.

The simplest sequential logic element: The set/reset (SR) latch

SR latch

- Latch constructed of cross-coupled NOR gates

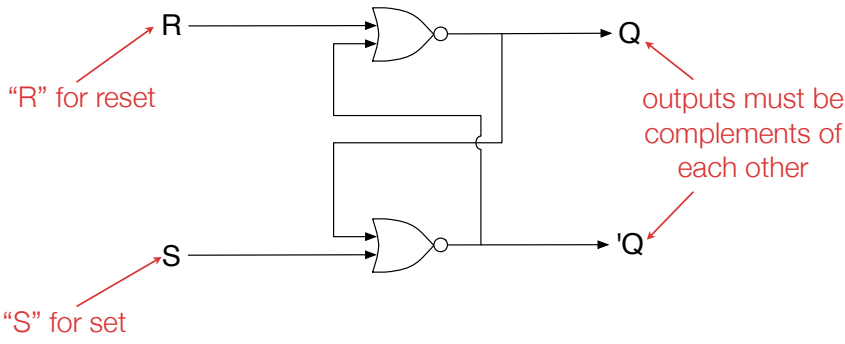
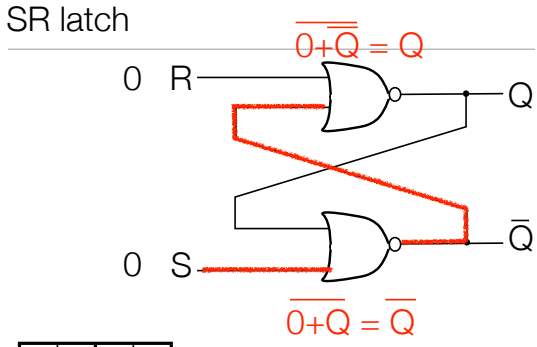


Figure: Source: Mano & Kime

The simplest sequential logic element: The set/reset (SR) latch



R	S	Q	$\overline{Q}$
0	0	Q	$\overline{Q}$
0	1	1	0
1	0	0	1
1	1		

Figure: Source: Mano & Kime

6 transistor SRAM cell

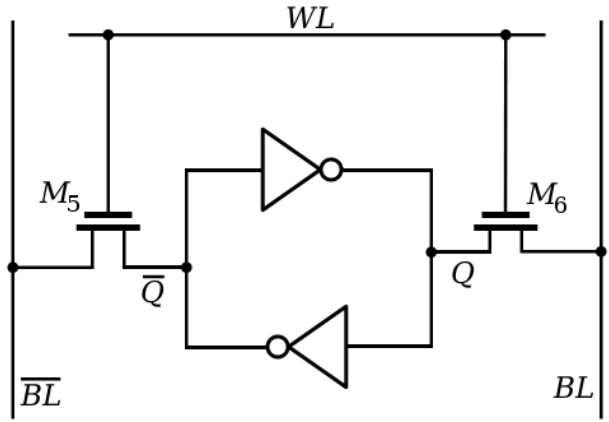
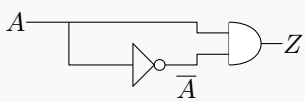


Figure: Source: Wikimedia

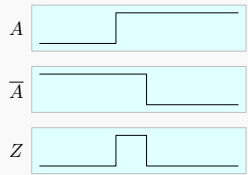
Asynchronous / Synchronous circuits

Timing

Circuit:



Voltages over time:



13