Free Hardware Design

Muhammadreza Haghiri

Who am I?

- Muhammadreza Haghiri
- Web : haghiri75.com (Persian)
  haghiri75.com/en (English)
- Telegram : t.me/hwtroll
- Also, you can find me on twitter and instagram : prpe26

Outline

- Why Free Designs?
- Examples
- Logical Design
- Digital Electronics
- HDL?
- Challenges
- Q & A

Why Free Designs?

- More eyes see what we’ve done
- People will trust us easier
- Our ideas grow faster
- No possible backdoors

Examples

- MIPS Processor
- ARM Processor
- Arduino
- Raspberry Pi
- And millions of other devices!

Logical Design

- It contains developing idea of a special purpose hardware,
- Construction of Logical functions and applications
- Simplifying functions

Logical Design

- As an example, consider a simple “one bit comparator”

A	B	A < B	A = B	A > B
0	0	0	1	0
0	1	1	0	0
1	0	0	0	1
1	1	0	1	0

Logical Design

- The functions are these :

  A = B : ~A ~B + AB
  A < B : ~A B
  A > B : A ~B

- We can’t simplify this, but of course we can simplify other devices. Let’s see.

Logical Design

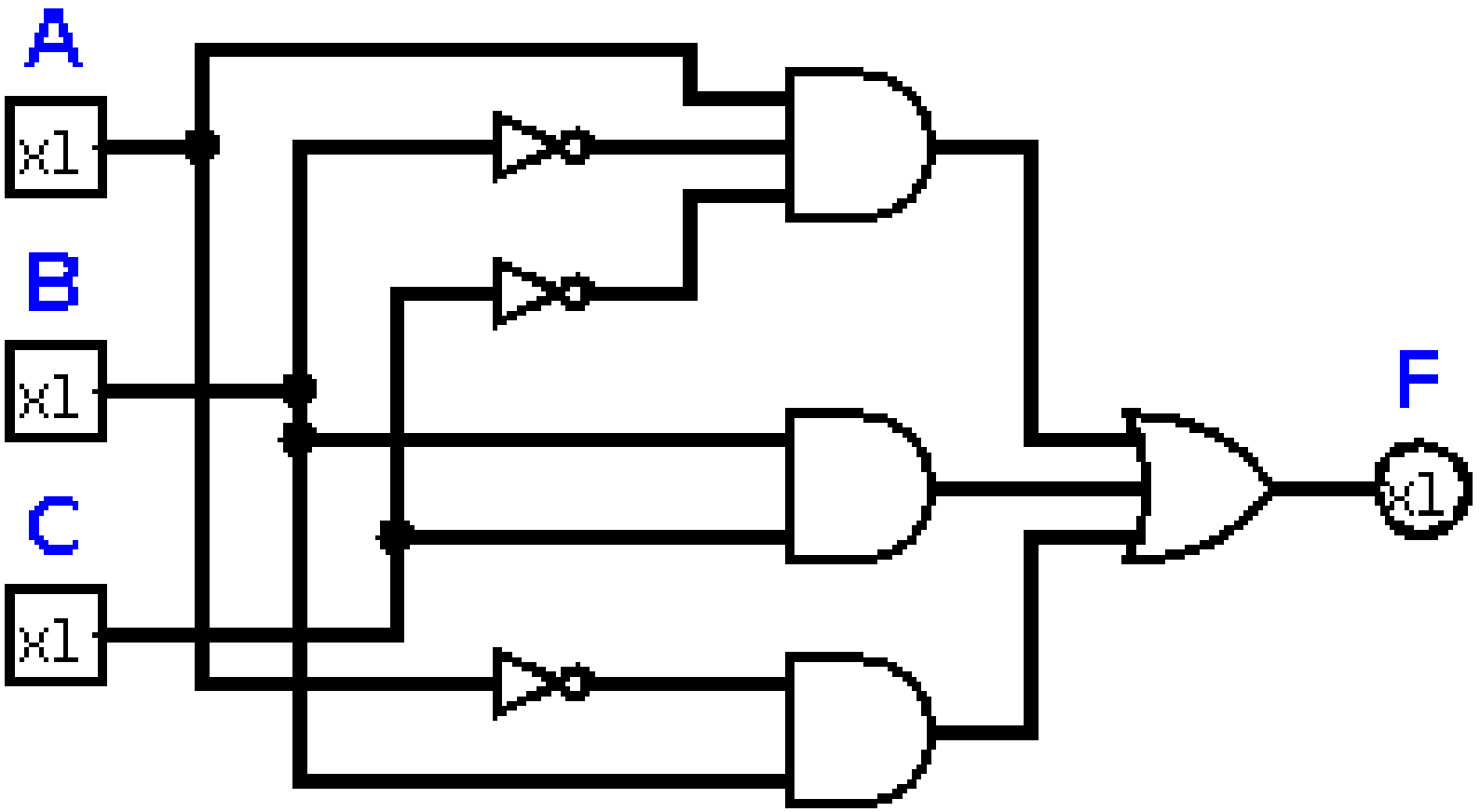
- Now, imagine a circuit with three inputs. We want our circtuit to be 1 on 2,3,4,7 conditions.

A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Logical Design

- When we want to write functions, it seems to be:
  ~A B ~C + ~A B C + A ~B ~C + A B C
- But, when we map it using Karnaugh map, it will become :
  A ~B ~C + BC + ~A B

Logical Design



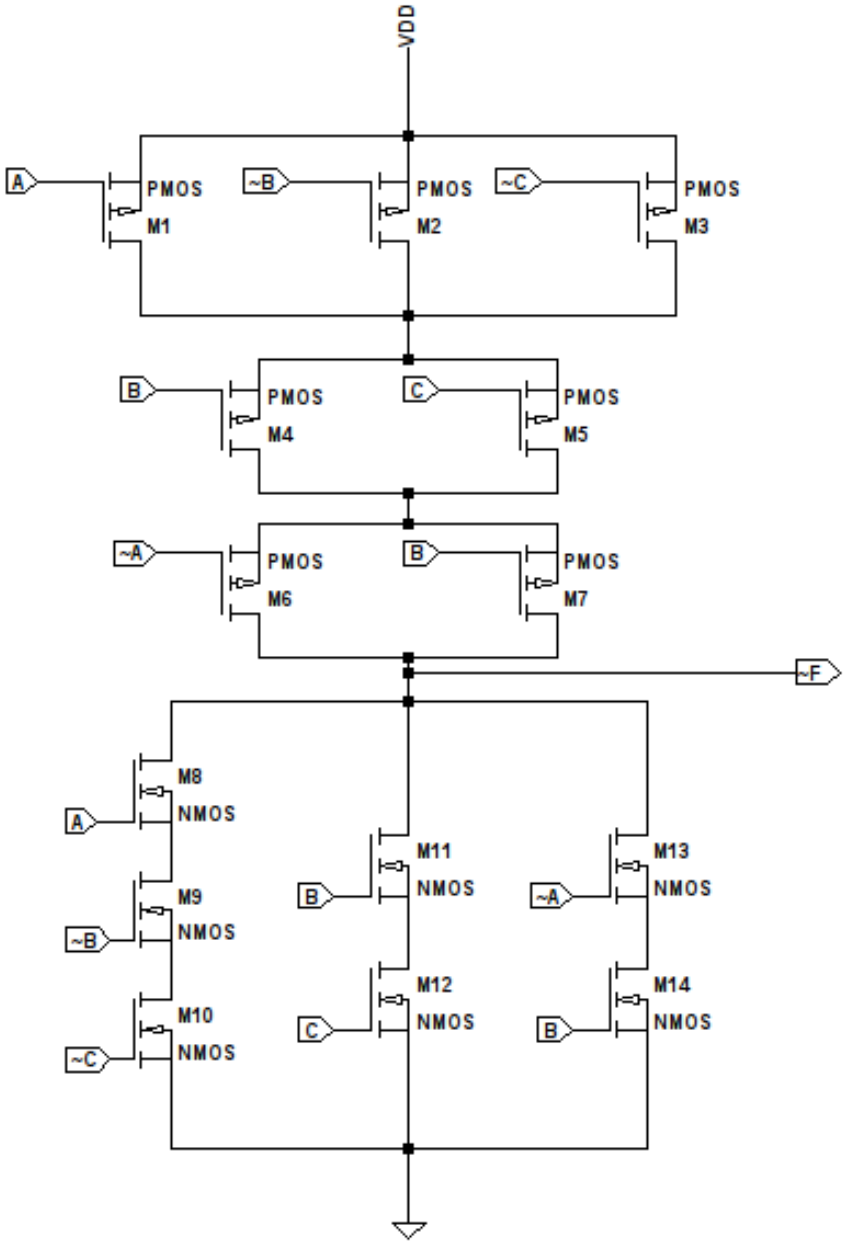
Digital Electronics

- Now, after we designed our logical circuit, we need to implement it using transistors. It’s called “Digital Electronics”.
- For this presentation, we use CMOS, or complementary metal oxide semiconductor.

Digital Electronics

- We needed simplifying, because …
- The digital circuit should be implemented with the minimum number of transistors
- Also, we need design inverters (We don’t draw them in the digital schematics!)

Digital Electronics




NOT MORE DIGITAL ELECTRONICS
TELL ME MORE ABOUT THE EASY WAY!

HDL?

- Stands for “Hardware Description Language”
- It’s used to program CPLD’s (Complex Programmable Logic Device) and FPGA’s (Field Programmable Gate Array)
- Cheaper than the digital electronics!

HDL?

- There are 10 most popular variations of HDL, VHDL and Verilog HDL.
- VHDL is similar to Ada (and 8086 assembly)
- Verilog is similar to our beloved C.

HDL?

- Do you remember our circuit? In next slide we'll see two implementation of that circuit, in both VHDL and Verilog!

HDL?

- VHDL

```
GATE_1 : OR_GATE_3_INPUTS
   GENERIC MAP ( BubblesMask                => 0)
   PORT MAP ( Input_1                       => s_LOGISIM_NET_7,
              Input_2                       => s_LOGISIM_NET_6,
              Input_3                       => s_LOGISIM_NET_8,
              Result                        => s_LOGISIM_NET_9);

GATE_2 : AND_GATE
   GENERIC MAP ( BubblesMask                => 0)
   PORT MAP ( Input_1                       => s_LOGISIM_NET_0,
              Input_2                       => s_LOGISIM_NET_1,
              Result                        => s_LOGISIM_NET_6);

GATE_3 : NOT_GATE
   PORT MAP ( Input_1                       => s_LOGISIM_NET_0,
              Result                        => s_LOGISIM_NET_4);

GATE_4 : AND_GATE
   GENERIC MAP ( BubblesMask                => 0)
   PORT MAP ( Input_1                       => s_LOGISIM_NET_5,
              Input_2                       => s_LOGISIM_NET_0,
              Result                        => s_LOGISIM_NET_8);

GATE_5 : NOT_GATE
   PORT MAP ( Input_1                       => s_LOGISIM_NET_2,
              Result                        => s_LOGISIM_NET_5);

GATE_6 : NOT_GATE
   PORT MAP ( Input_1                       => s_LOGISIM_NET_1,
              Result                        => s_LOGISIM_NET_3);

GATE_7 : AND_GATE_3_INPUTS
   GENERIC MAP ( BubblesMask                => 0)
   PORT MAP ( Input_1                       => s_LOGISIM_NET_2,
              Input_2                       => s_LOGISIM_NET_4,
```

- Verilog

```
/*************************************************************************
 ** Here all input connections are defined                              **
 *************************************************************************/
assign s_LOGISIM_NET_1                  = C;
assign s_LOGISIM_NET_2                  = A;
assign s_LOGISIM_NET_0                  = B;

/*************************************************************************
 ** Here all output connections are defined                             **
 *************************************************************************/
assign F                                = s_LOGISIM_NET_9;

/*************************************************************************
 ** Here all normal components are defined                              **
 *************************************************************************/
OR_GATE_3_INPUTS #(.BubblesMask(0))
   GATE_1 (.Input_1(s_LOGISIM_NET_7),
           .Input_2(s_LOGISIM_NET_6),
           .Input_3(s_LOGISIM_NET_8),
           .Result(s_LOGISIM_NET_9));

AND_GATE #(.BubblesMask(0))
   GATE_2 (.Input_1(s_LOGISIM_NET_0),
           .Input_2(s_LOGISIM_NET_1),
           .Result(s_LOGISIM_NET_6));

NOT_GATE      GATE_3 (.Input_1(s_LOGISIM_NET_0),
                      .Result(s_LOGISIM_NET_4));

AND_GATE #(.BubblesMask(0))
   GATE_4 (.Input_1(s_LOGISIM_NET_5),
           .Input_2(s_LOGISIM_NET_0),
```


CHALLENGES!
CHALLENGES EVERYWHERE...
makeameme.org

Challenges

- In logical design, make sure your design produces expected results.
- In digital designs, you need to test your designs for delay, area, power and corners. This is important!
- Also, the final product shall be tested before release and marketing!

Challenges

- Important challenge is that you may need to redesign some parts even after the final production and fabrication. To prevent this, double check and triple check your logical and digital designs!


DEAR LORD!
THEY STILL THINK I
SHALL REPAIR THEIR PC!

Q & A