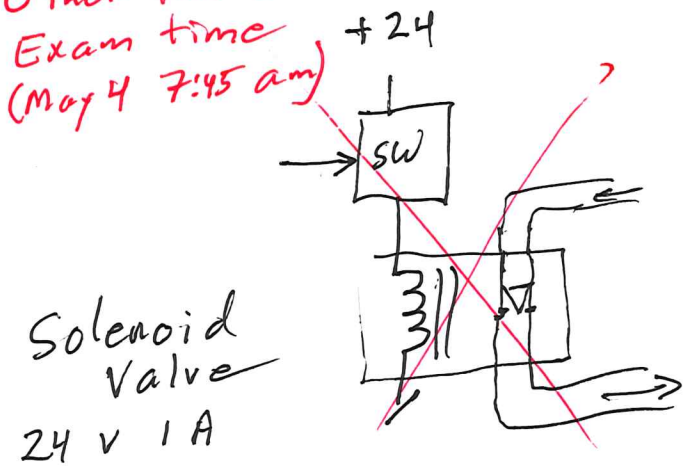
Ph 623 - 16 April 20

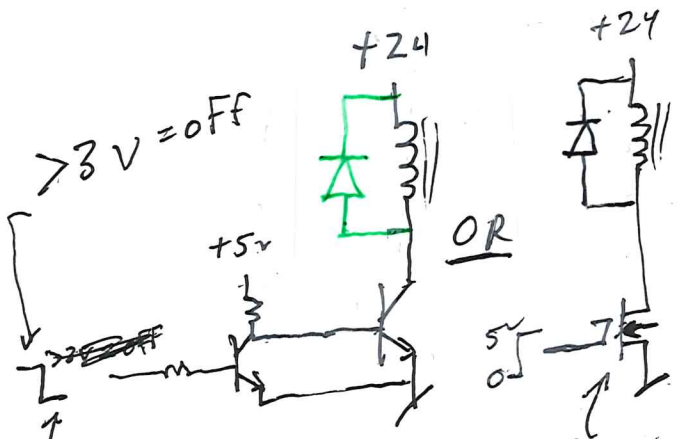
- HW 10 due next Thursday
  - PLL Prelab
- No A/D or PLL Lab
  - We will do 3 Xilinx labs
  - First one Fixed instructions
  - Other two open-ended - make whatever you like.
- Read H + H
  - Short section 10.5.4 on p 745
  - 764 - 777 (11.1 - 11.3.4) can skip ABEL
  - 11.3.5 = micro controllers } optional
  - Also all of Ch 15 }
- Ch 12 - interfacing } reference

Today

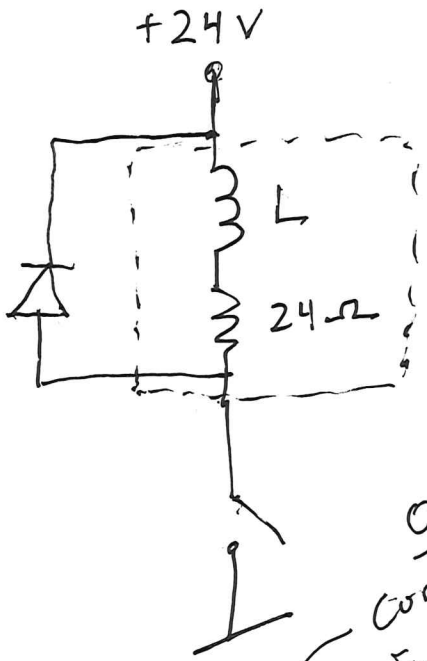
- Programmable Logic
- A little interfacing

- FPGA Software Working ?
- FPGA I due < Monday 27th
- Other two Labs due at Final Exam time (May 4 7:45 am)

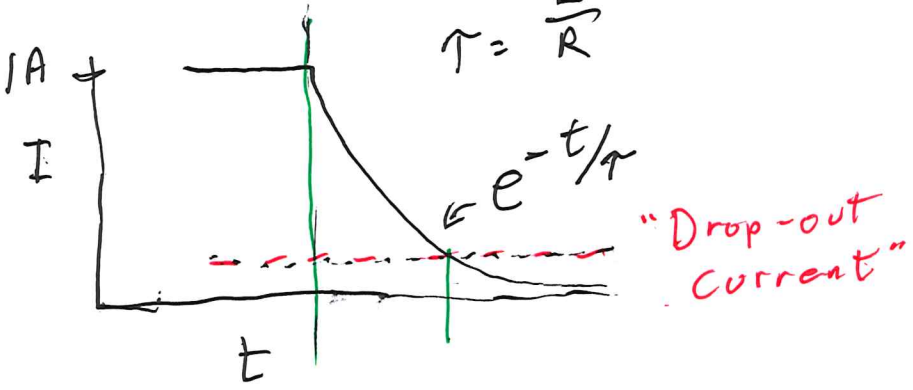




Slow Shutoff



Open sw
Current transfers from switch to diode

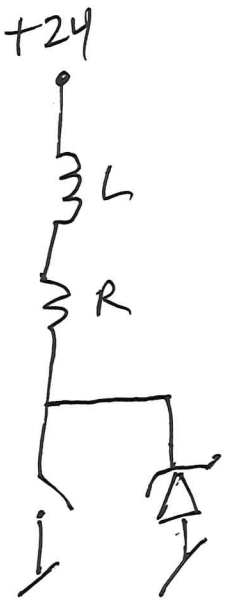


$\tau = \frac{L}{R}$

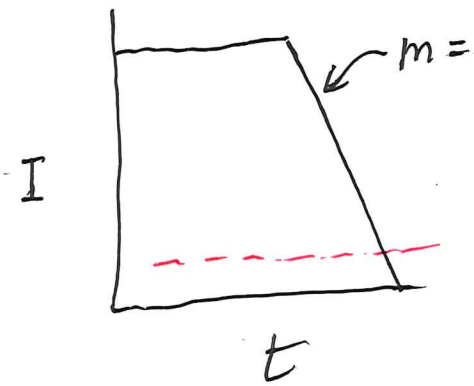
$e^{-t/\tau}$

"Drop-out Current"

- FIX

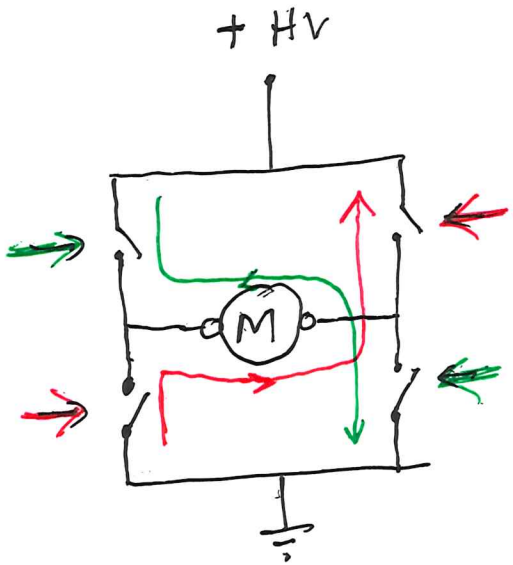


Zener
$V_Z < V_{MAX}$ Transistor
$I_{MAX} > I_{SOLENOID}$

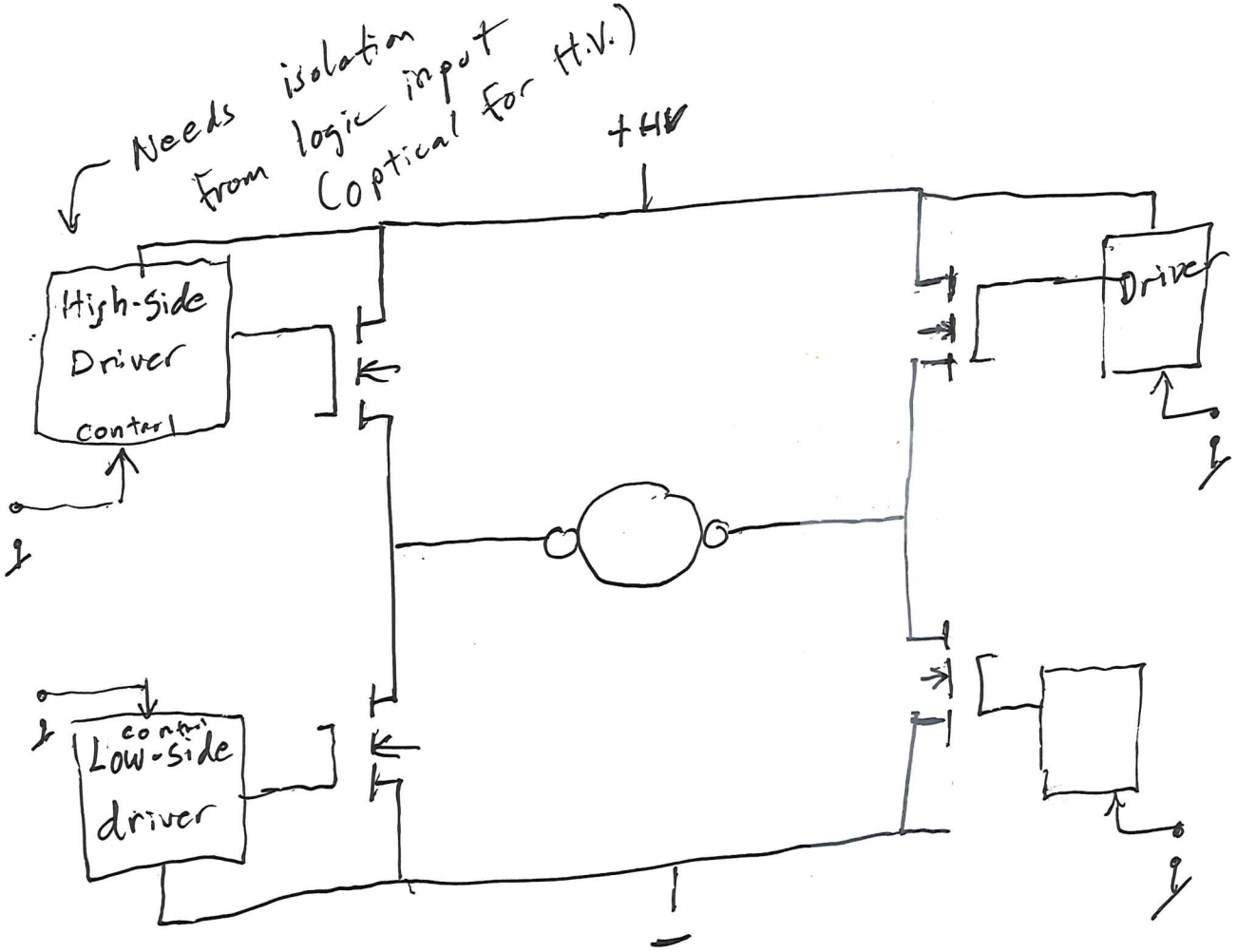


$m = -\frac{V_Z (+IR)}{L}$

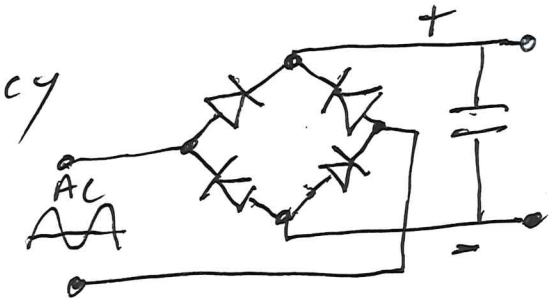
Reversible Load



"H-Bridge"



Also high-efficiency Substitute For



= "Schottky Diode" (metal → semiconductor) ~0.3 V in Si

MOSFET Better (.01 Ω)

Types of Programmable Logic

PAL
PLD > Don't Worry about it

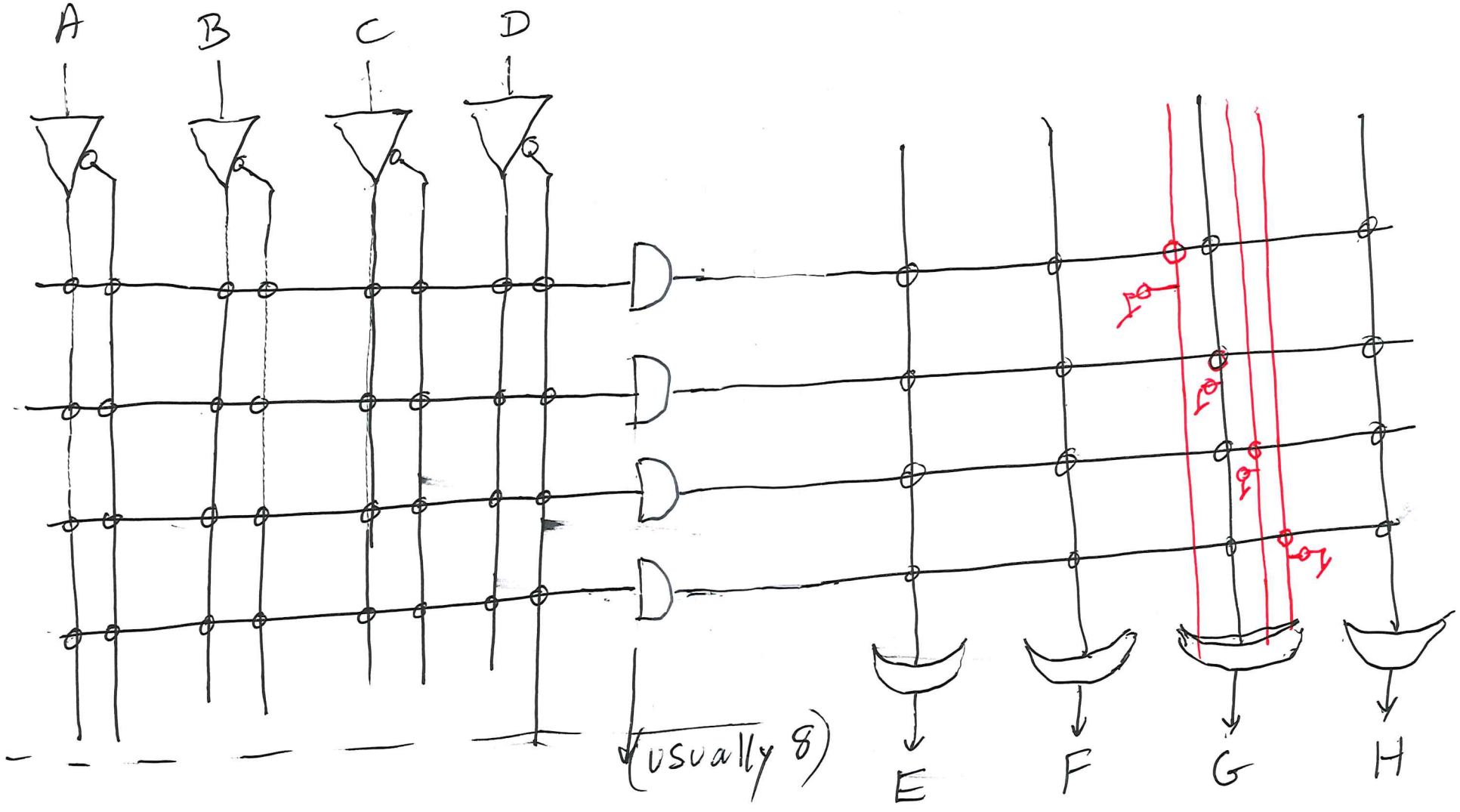
CPLD - Just big PLD "Complex"

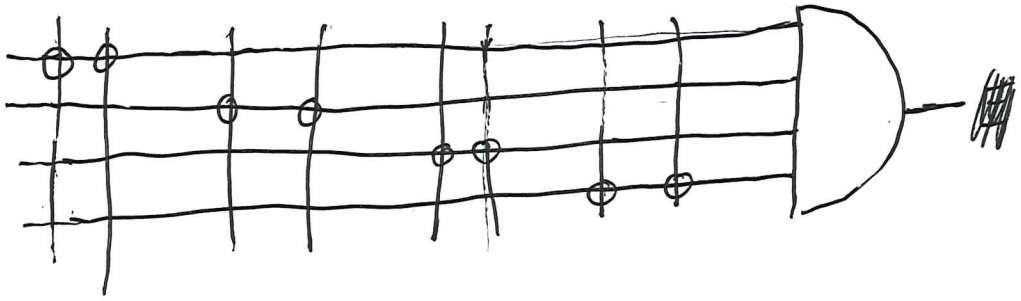
FPGA

"Field Programmable Gate Array

- $10^4 - 10^7$ logic elements
  (DE0-nano $\sim 3 \times 10^4$)
- 10 MHz - 10 GHz
  (DE0-nano = 100 MHz)
- Different organization of logic
  - LUTs + F/F

CPLD "Unit Cell"

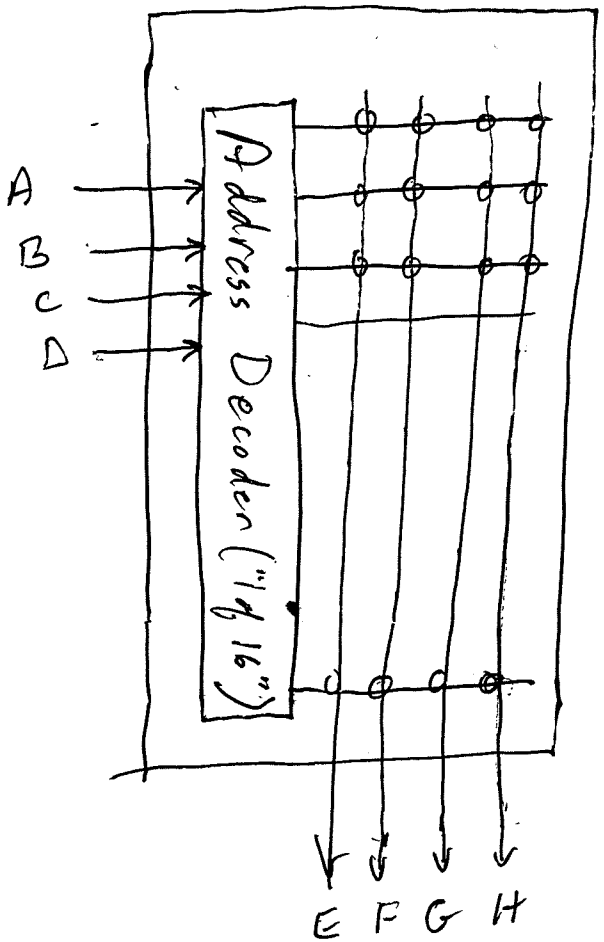


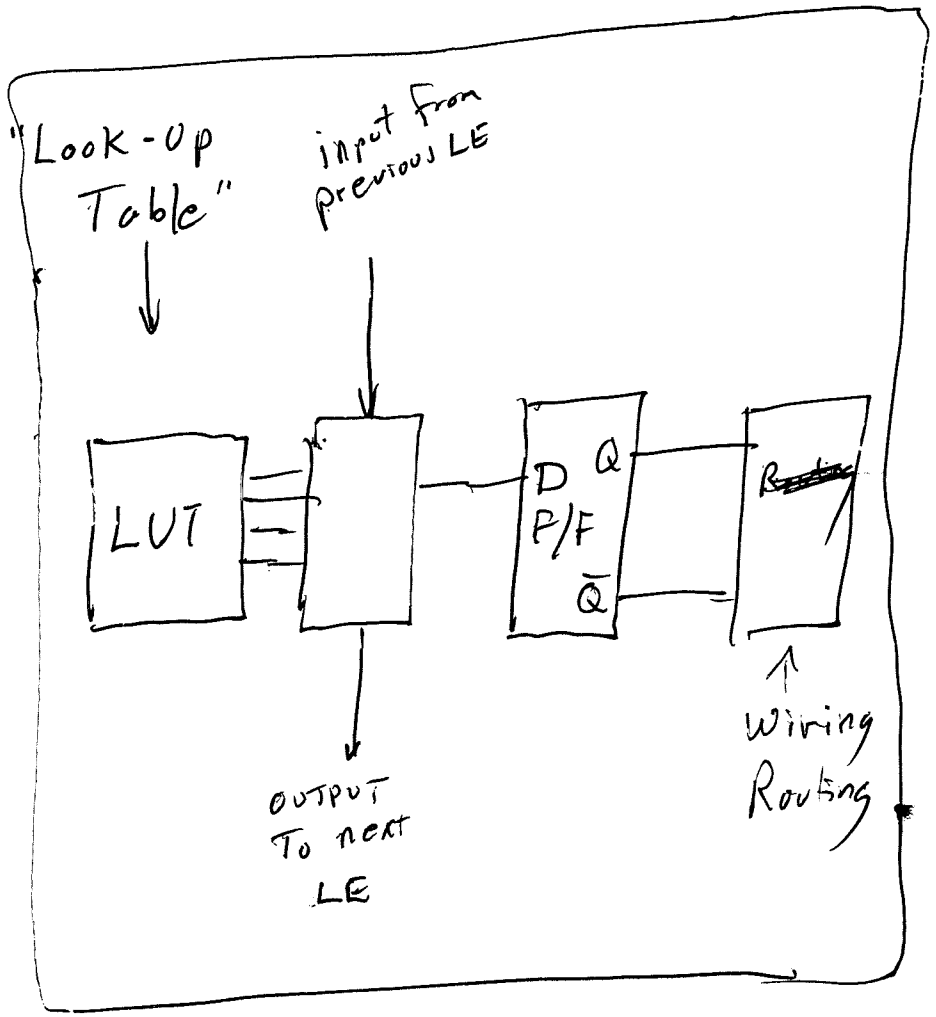


Can implement any 4-input 4-output Truth Table with 8 lines

FPGA Architecture

Read Only Memory





"Logic Element"

L.E.

DE0-nano has ~30,000 Organized into larger arrays

Fancier FPGAs

Even DE0 nano is big enough you can program in a whole computer. P.C. equivalent for DE0 is called Nios II - They'll provide a software unit you can add to your program — use rest of LE's to implement "Glue Logic" on input + output. These software units are called "Intellectual Property". Also available for: USB interfacing, Ethernet interfacing, Disk drive " } search web

Often have to pay for I.P. (Nios II is free)

Larger "Hybrid" FPGAs have some of these things hardwired (faster + more efficient)
e.g.: Four PC's + four Ethernet + HDMI + USB + lots of LE's left over

6 1/2

Programming

on-chip memory for each of the switches

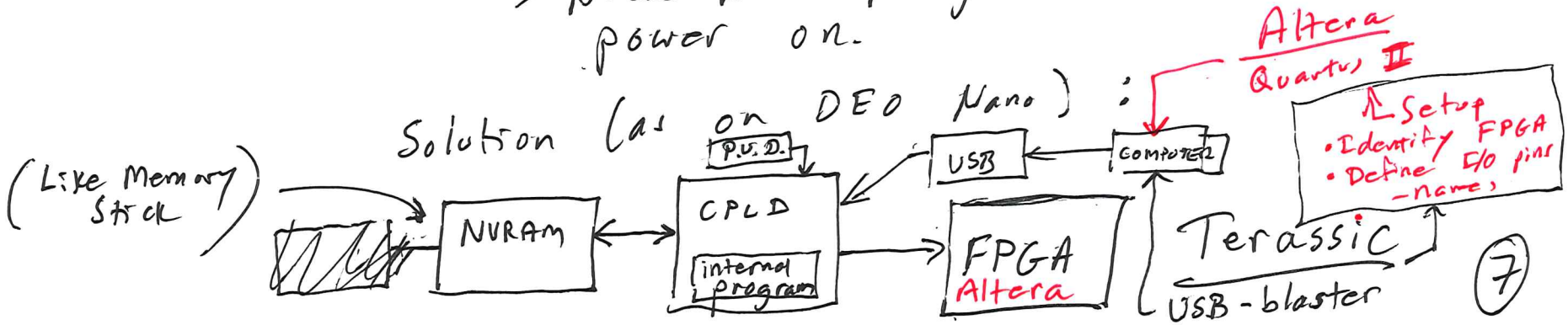
CPLD — Usually non-volatile "EEPROM" (Electrically-Erasable — replaced "UV-erasable")

Usually requires higher than standard voltage, so use "POD" to program — ~~USB~~ Computer → USB → POD → CPLD

FPGA — Chip processing in compatible with EEPROM in most cases. Uses volatile CMOS RAM.

⇒ Need to reprogram after each power on.

Solution (as on DE0 Nano):



How to Program The FPGA

Altera Software ("Quartus - II")

Hides internal structure
Just tell it what you want the device to do!

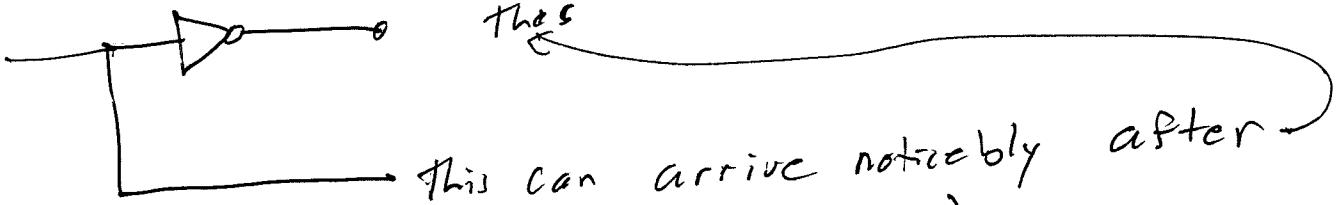
Two ways:

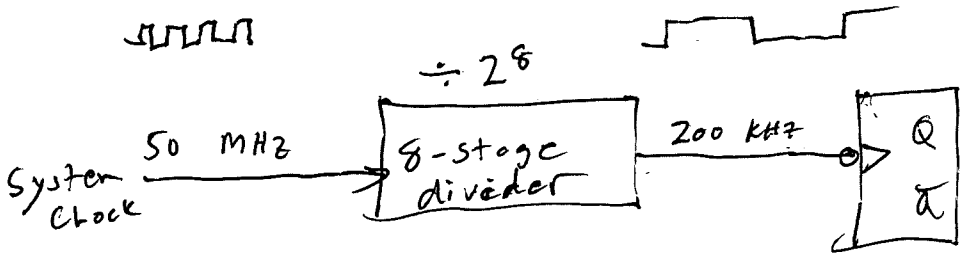
1. Schematic Capture
   Library of familiar digital ICs
   AND, OR, NOT, XOR
   J/K, D
   Multiplexers, Data Latches
   (Data Sheets)

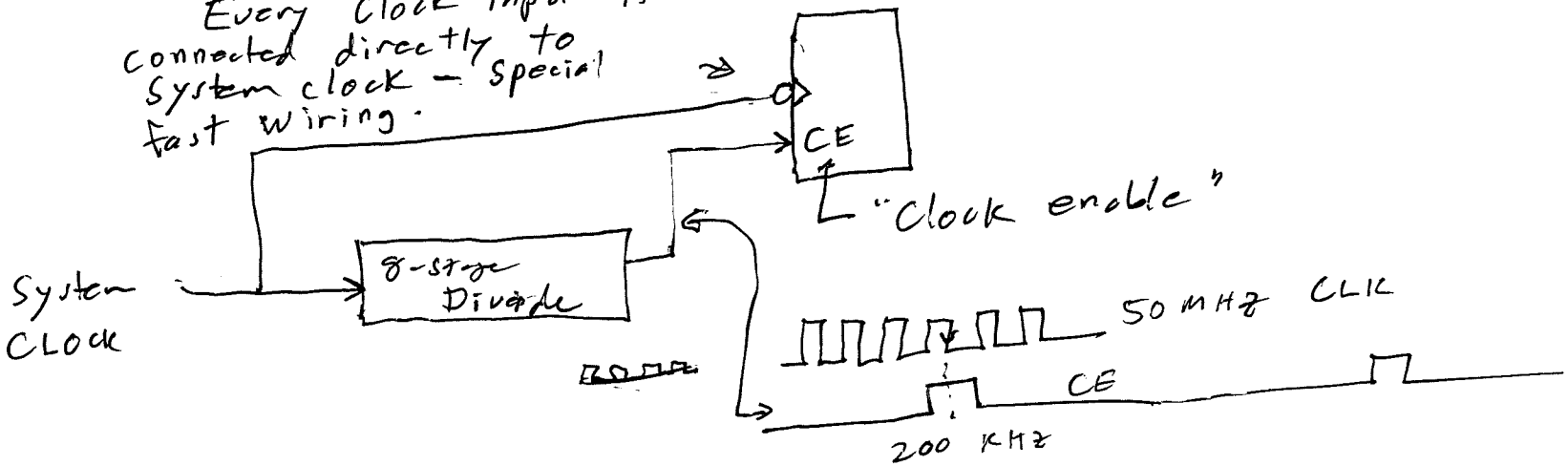
2. "Hardware Descriptor Languages" HDLs
   a) Verilog - more like C { Functional (- efficiency)
                            { Descriptive
   b) VHDL - more strongly typed
      modeled on ADA
      More verbose - higher level
      Europe - Defense
      Descriptors

Caveat (Particularly for Schematic entry)

Fast logic
Slow wires (may go through many switches)



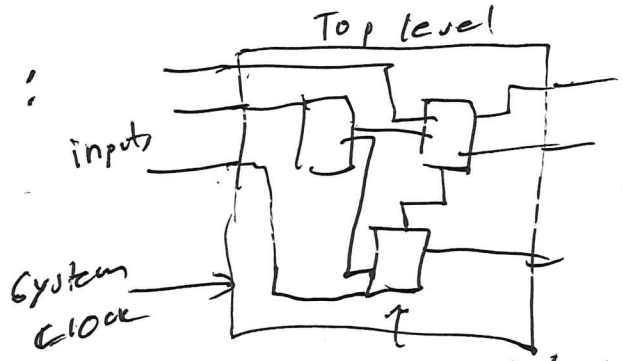




Quartus II

1) Accepts Schematic or HDL program

Hierarchical:



sub-modules Described separately

Can mix Schematic + HDL

Top level usually schematic

2) Checks for errors

3) Can Simulate (catch and diagnose timing errors)

4) "Compiles" into program For FPGA