Combinational Logic; Hierarchical Design and Analysis

Tom Kelliher, CS 240

Feb. 10, 2012

1 Administrivia

Announcements

Collect assignment.

Assignment

Read 3.3.

From Last Time

IC technology.

Outline

1. Combinational logic.
2. Hierarchical design
3. Design analysis.

Coming Up

Design example.

2 Combinational Logic

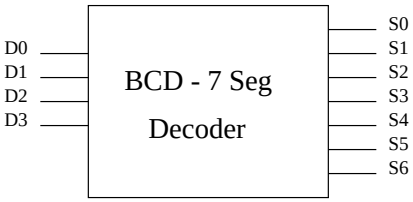
1. Definition: Logic circuits in which the output(s) depend solely upon current inputs.

2. No feedback or memory.

3. Sequential circuits: outputs depend upon current inputs *and* previous inputs.

   Memory or registers.

4. Example — BCD to 7-segment decoder:



3 Hierarchical Design

1. Transistor counts:

Processor	Year	Transistor Count
TI SN7400	1966	16
Intel 4004	1971	2,300
Intel 8085	1976	6,500
Intel 8088	1979	29,000
Intel 80386	1985	275,000
Intel Pentium	1993	3,100,000
Intel Pentium 4	2000	42,000,000
AMD Athlon 64	2003	105,900,000
Intel Core 2 Duo	2006	291,000,000
Intel Core 2 Quad	2006	582,000,000
NVIDIA G80	2006	681,000,000
Intel Dual Core Itanium 2	2006	1,700,000,000
Intel Atom	2008	42,000,000
Six Core Xeon 7400	2008	1,900,000,000
AMD RV770	2008	956,000,000
NVIDIA GT200	2008	1,400,000,000
Eight Core Xeon Nehalem-EX	2010	2,300,000,000
10 Core Xeon Westmere-EX	2011	2,600,000,000
AMD Cayman	2010	2,640,000,000
NVIDIA GF100	2010	3,000,000,000
Altera Stratix V	2011	3,800,000,000

2. Design and conquer:

   CPU ⇒ Integer Unit ⇒ Adder ⇒ binary full adder ⇒ NAND gates

3. Reuse:

   Once logic is collected into a block, it can be instantiated several times in several places.

   Adders are used at several points within a CPU: integer ALU, program counter circuit, multiplier, etc.

   Binary full adders are connected to form adders.

4. Scaling:

   Consider the two-dimensional tiling of memory cells.

These techniques reduce the number of transistors which must be laid out "by hand."

Design styles:

1. Top-Down design: divide and conquer.

2. Bottom-Up design: promotes reuse.

The savvy designer often uses both techniques within a single project.

Design tools:

1. CAD tools: programs to assist with schematic capture, HDL entry, synthesis, simulation. Running on "first silicon."

   The "old" days: drafting tables, taping out a circuit, and lots of prototyping.

2. HDLs and synthesis

   (a) Why VHDL is my favorite acronym.

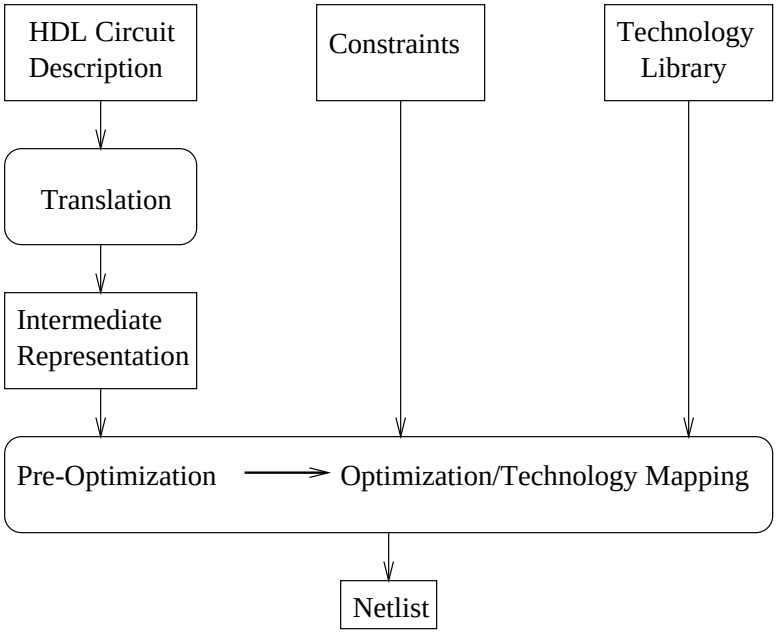
   (b) What is VHDL? 4-1 mux example:

```
library ieee;
use ieee.std_logic_1164.all;

entity mux4_1 is
   port (a0: in bit;
         a1: in bit;
         d0: in bit;
         d1: in bit;
         d2: in bit;
         d3: in bit;
         z: out bit);
end mux4_1;

architecture behavioral of mux4_1 is
signal address: bit_vector(1 downto 0);
begin
   address <= a1 & a0;
   with address select
      z <= d0 when "00",
           d1 when "01",
           d2 when "10",
           d3 when "11";
end behavioral;
```

(c) Synthesis process:



4 Design Analysis

Combinational circuit analysis — "reverse engineering." Skip.

Logic Simulation:

1. Vital today: First silicon must run.

   Can't re-wire a die.

   Entire computers have been simulated to the point of booting the OS.

2. Simulator used to verify circuit behavior and timing.

   Results are only as good as the tests run.

   Large circuits cannot be simulated completely. Just ask Intel (`fdiv`).

3. Netlist used to describe circuit. Text file.

4. Schematic: graphical representation of circuit. Tool to convert to netlist form.

5. User produces "test vectors," which are the inputs to the simulator.

   Good test vectors are the key to meaningful results.