

EECS 303: Advanced Digital Logic Design  
Final Exam

Robert Dick

6 December 2006

Show your work. Derivations are required for credit; end results are insufficient. Read each sentence.  
In some cases I indicate something that can save you a lot of work.

## 1 (10 pts.) VHDL

Consider the following VHDL code.

---

```
entity RECOG is
  port (
    clk, a, reset: in bit;
    h: out bit
  );
end RECOG;

architecture STATE_MACHINE of RECOG is
  type state_type is (s0, s1, s2, s3);
  signal ps, ns : state_type;
begin

  STATE: process (reset, clk)
  begin
    if (reset = '1') then
      ps <= s0;
    elsif (clk'event and clk = '1') then
      ps <= ns;
    end if;
  end process STATE;

  NEW_STATE: process (ps, a)
  begin
    case ps is
      when s0 =>
        case a is
          when '0' => ns <= s0;
          when '1' => ns <= s1;
        end case;
      when s1 =>
        case a is
          when '0' => ns <= s2;
          when '1' => ns <= s1;
        end case;
      when s2 =>
        case a is
          when '0' => ns <= s0;
          when '1' => ns <= s3;
        end case;
      when s3 =>
        case a is
          when '0' => ns <= s3;
          when '1' => ns <= s3;
        end case;
    end case;
  end process NEW_STATE;

  OUTPUT: process (ps)
  begin
    case ps is
      when s0 => h <= '0';
      when s1 => h <= '0';
      when s2 => h <= '0';
      when s3 => h <= '1';
    end case;
  end process OUTPUT;
end STATE_MACHINE;
```

---

1. Derive the corresponding state table.
2. Derive the corresponding state diagram.
3. Write the regular expression for the accepted sequences.

## 2 (10 pts.) State minimization

Minimize the following finite state machine. Show each step of minimization.

| Current State | Next State (input) |   | Output |
|---------------|--------------------|---|--------|
|               | 0                  | 1 |        |
| A             | D                  | C | 0      |
| B             | B                  | B | X      |
| C             | B                  | A | 1      |
| D             | D                  | A | 0      |
| E             | E                  | A | X      |

### 3 (10 pts.) Asynchronous finite state machine design

Codes will be supplied to your machine via two binary inputs,  $\alpha$  and  $\beta$ . The inputs will never change simultaneously. The combined pair of inputs permit four symbols to be entered,  $f = \overline{\alpha}\overline{\beta}$ ,  $g = \overline{\alpha}\beta$ ,  $h = \alpha\overline{\beta}$ , and  $i = \alpha\beta$ . The input is initially  $f$ .

1. Design an asynchronous finite state machine that will accept the following sequences:

$$f^+g^+f^+h^+i(f+g+h+i)^*$$

You need only carry out those stages of the design process up to and including state assignment and derivation of a fully-specified state table.

2. In three or fewer sentences, explain why there were so many plusses in the regular expression, e.g.,  $f^+$ .
3. In three or fewer sentences, explain why it would or would not be reasonable to use kernel extraction to minimize the state variable functions of this machine.

### 4 (10 pts.) Arithmetic

Show the circuit-level diagram of an adder that accepts two two-bit unsigned integers and produces a three-bit unsigned integer. There must be at most two levels of logic between the inputs and outputs.

### 5 (10 pts.) Hazards

Consider the following function:

$$f(a, b, c) = \sum(0, 1, 2) \tag{1}$$

Assume access to complemented and uncomplemented input literals.

1. Show the schematic for a hazard-free two-level CMOS implementation with a minimal number of transistors.
2. How many transistors does your implementation have?

### 6 (10 pts.) State machines

Either prove that the following two machines are equivalent or prove that they are different. For each machine, A is the start state. Two machines are equivalent if there exists no single sequence of inputs that, if applied to both machines, will produce a different sequence of outputs.

| Current State | Next State (input) |   | Output | Current State | Next State (input) |   | Output |
|---------------|--------------------|---|--------|---------------|--------------------|---|--------|
|               | 0                  | 1 |        |               | 0                  | 1 |        |
| A             | D                  | E | 0      | A             | B                  | D | 0      |
| B             | A                  | E | 1      | B             | D                  | A | 1      |
| C             | A                  | E | 0      | C             | F                  | D | 0      |
| D             | C                  | A | 1      | D             | E                  | D | 0      |
| E             | A                  | C | 0      | E             | F                  | D | 0      |
| F             | A                  | C | 1      | F             | D                  | C | 1      |

## 7 (0 pts.) Research opportunity

1. How many unique and minimal deterministic finite state machines of  $l$  input symbols,  $m$  output symbols, and  $n$  states exist?
2. Explain how to design an asynchronous self-correcting finite state machine, i.e., a finite state machine that will have some specified behavior even if the indicated transition will be followed with probability  $\alpha$  where  $\alpha > 0.5$ .
3. Describe the simplest metric capable of accurately quantifying the exploitable memory access patterns in a heterogeneous NUMA processor.