Name Last Name ID Signature

EE 445 Computer Architecture I, Short Exam 1
October 24, 2024
Number of questions: 2. **Time allowed: 50 minutes.**

This exam only covers Algorithmic State Machines and Register Transfer Language (plus the relevant EE 348 Review).

Please show your work. Only what you write on the solution sheets will be graded. Best of luck.

Question 1 (25 points):

An imply microoperation is defined for registers A and B as follows:

- If bit B_i is 0, bit A_i is complemented.
- If bit B_i is 1, bit A_i is set to 1.

Example:

A(t):	1 1 0 0
B:	1 0 1 0
A(t+1):	1 0 1 1

(a) (8 points) What is the RTL representation for this micro-operation (put your Boolean expression in the simplest form)?

(b) (17 points) Design and draw the typical cell A_i of a register A implemented by an SR flip flop (FF). Remember that for an SR FF, $Q(t+1) = S+R'Q(t)$, where $SR=0$. The control signals and the RTL microoperations are as follows (recall that logic microoperations are always bit-wise):

P_1: Imply (as described above) using register B.
P_2: $A \leftarrow A \wedge B$ (here $\wedge$: LOGICAL AND).

Use the **separate design method**. At most one of the control signals can be asserted at a time. If none of the control signals P_1, P_2 are asserted, A keeps its value. In your drawing, clearly show the typical register cell A_i with its SR FF, the required Boolean expressions (combining control signals along with A_i, B_i), and the clock. No need to draw the gates for these Boolean expressions; simply write down the expressions where needed in the drawing.

Question 2 (75 points):

In this question, you are required to design a **locker password-checking security system**, which allows a user up to three attempts only to enter the correct password so that the locker will be opened. You have the following hardware available for the system design:

- Three signals *S*, *Match*, and *Stop*.
- Three flip flops (FFs) *Allow*, *Safe*, and *Unsafe*.
- Two n-bit registers *Ref* and *User*.
- A combinational n-bit adder-subtractor, and its n-bit output signal is *A*.
- A synchronous 2-bit counter *C*.

The locker password-checking security system operates on numeral passwords (just numbers) as follows:

- The system has an initial idle state.
- The start signal *S* is set and cleared by the user, while the FF *Allow* is set externally via parental administration (out of your system).
- When the start signal *S* is set to 1 and if the FF *Allow* is 1, the system clears FFs *Safe* and *Unsafe*, loads the counter *C* with $(01)_2$, and loads the reference password into *Ref*.
- On the next clock cycle, the system loads the user-entered password into *User*.
- On the next clock cycle, the system compares the two passwords via the adder-subtractor, which takes its two inputs from *Ref* and *User*.
- If the two passwords match (*Match* signal created from *A* is 1), the FF *Safe* is set to 1 (so that the locker will be opened), and the system returns to the idle state.
- If the two passwords do not match (*Match* signal is 0), the FF *Unsafe* is set to 1 (so that a warning will be issued to the user), and the counter *C* is incremented.
- On the next clock cycle, the counter *C* value is checked.
- If *C* is all zeros (checked via *Stop* signal created from *C*), the FF *Allow* is cleared to 0, and the system returns to the idle state. This means the user is temporarily blocked, and the system will not leave the idle state unless both *S* and *Allow* are 1's.
- If *C* is not all zeros, the FF *Unsafe* is cleared to 0, and the system returns to the state of allowing the user to re-enter the password.
- You do not need to worry about opening the locker nor issuing the warning.

(a) (25 points) Derive an ASM chart (your ASM algorithm) representing a circuit that executes the procedure of this locker password-checking security system **using exactly four states** (including the idle state). Specify all the states. Clearly indicate the register operations with proper RTL syntax. Specify where the signal *A* is updated on the ASM chart.

(b) (25 points) Draw a block diagram of the data path unit with all the involved registers, flip flops, as well as synchronous and combinational components. Specify all the associated signals.

(c) (25 points) Design and draw the controller unit using the multiplexer and D FF method.