University of California at Berkeley
College of Engineering
Department of Electrical Engineering and Computer Science

EECS 150 R. H. Katz
Fall 2000

FINAL EXAMINATION
Thursday, 14 December 2000

INSTRUCTIONS—READ THEM NOW! All work is to be done in your blue books. Print and sign your name and Student ID Number on the cover of each book you use. Partial credit is given only if we can evaluate your approach: indicate your assumptions and ***write as neatly*** as possible. The examination is worth 150 points. You have three hours to complete your work. Spend 20-30 minutes thinking about the problem specification before you commence. This is an open note/open book examination.

It is a sad fact of life that cheating happens. It will not be tolerated. By signing your blue book, you assert that all of the work included therein is your own, and that you understand the harsh penalties that will be imposed should cheating be detected—a 0 on the examination, and a letter of reprimand to your file:

QUESTION	POINTS ASSIGNED
1	20
2	40
3	20
4	20
5	50
TOTAL	150

Design Specification: READ CAREFULLY! Please send 20-30 minutes studying the specification.

Your task is to design the control state machine and datapath for a programmable video cassette recorder to the following specification:

- The VCR has the ability to store up to two programs at a time, PROGRAM[0] and PROGRAM[1].

- Each program is described by the triple <Channel, Week, StartTime, EndTime>.
  StartTime is <Day of Week, Time of Day> and EndTime is <Day of Week, Time of Day>, where:
  Channel (CH): $0 < CH \leq 99$;
  Week (WK): 0 = the current week, 1 = next week;
  Note: a new week begins when Saturday wraps around to Sunday;
  Day of the Week (DoW): Su (0), M (1), Tu (2), W (3), Th (4), F (5), Sa (6);
  Time of Day (ToD): HH:MM AM/PM where $00 < HH \leq 12$, $00 \leq MM \leq 59$, AM(0)/PM(1);
  Note that Noon is 12:00 PM and Midnight is 12:00 AM;

- By default, at power up, the DoW is set to 0 (Sunday) and the ToD to Noon (12:00 PM). Thus the user must program the VCR with the correct current DoW and ToD.

The controller has two modes, SET (0) and RUN (1).

- When SET is asserted, the user console *described on the next page* allows the user to do any or all of the following three actions: (1) set the current DoW/ToD, (2) set PROGRAM[0], or (3) set PROGRAM[1]. Exiting SET mode places the controller into RUN mode.

- When RUN is asserted, the VCR has two more modes: REC (0) and PLAY (1). When REC is asserted, recording will commence when the current DoW/ToD is equal to one of the two programs' StartTime and WK indicates that it is the current week. Recording stops when the current DoW/ToD is equal to the EndTime (with WK=0) of the program that caused recording to commence.

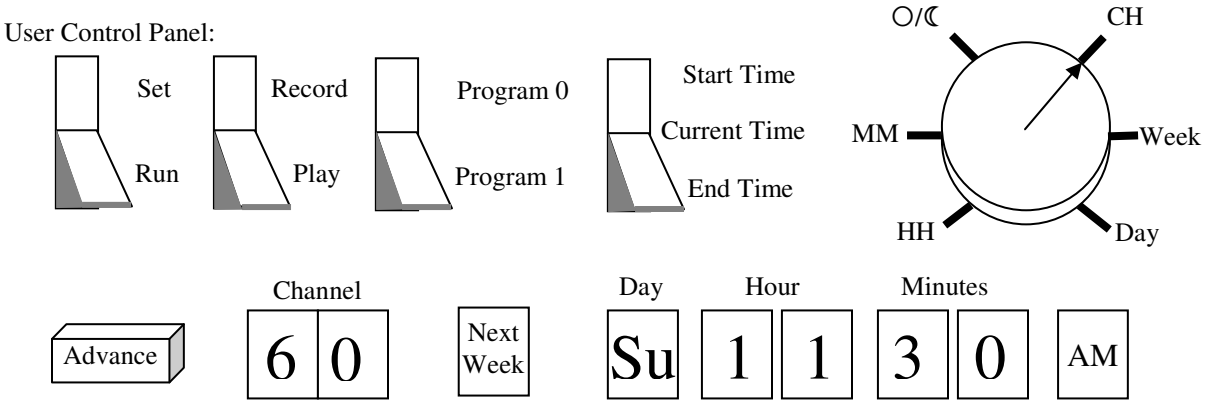
- ***For the purposes of this examination, you will not need to design the details of the PLAY mode, just REC mode.***

- When in SET mode or RUN/PLAY mode, recording will be inhibited even if the current time matches a program start time. If the VCR is currently recording, and the user exits RUN mode to enter SET mode, the current recording is immediately terminated.

- Since the VCR can only record one program at a time, overlapping programs are ignored. This is not quite the same thing as Don't Care! The earlier start time has precedence, and *the recording will continue through to its matched end time*. In other wordsm once recording commences, only the EndTime of the program that initiated the recording is checked.

- PROGRAM[0] and PROGRAM[1] are initially invalid (that is, unprogrammed). They become valid once they are set by the user. After a taping session has finished, the program becomes invalid, and will not be able to be used again until it is reprogrammed by the user. If a program was not activated for some reason, for example, it overlapped with another program with an earlier start time, it remains valid and could cause taping to commence in a future week.

- Thus, recording can only take place if there is a valid program, the current time matches that program's time start, and the VCR is in REC mode.



The user console works in the following way. SET/RUN, REC/PLAY, Program 0/Program 1, and StartTime/Current Time/EndTime are set via "rocker" switches. The first three operate in two positions (top in, bottom in), the fourth can be set in three positions (top in/bottom in/in between). With the switch's top pushed in, the top signal is indicated (e.g., Set, Record, Program0, StartTime). When the bottom is pushed in, the lower signal is indicated (e.g., Run, Play, Program1, EndTime). In the case of the last switch, the "in between" setting indicates that the current time is to be set.

There is also a six position rotary switch, which is used to indicated one of CH, This/Next Week, Day, HH (hour), MM (minutes), and morning/evening (AM/PM). This is used to set the current time, and the StartTime and EndTime of the two programs.

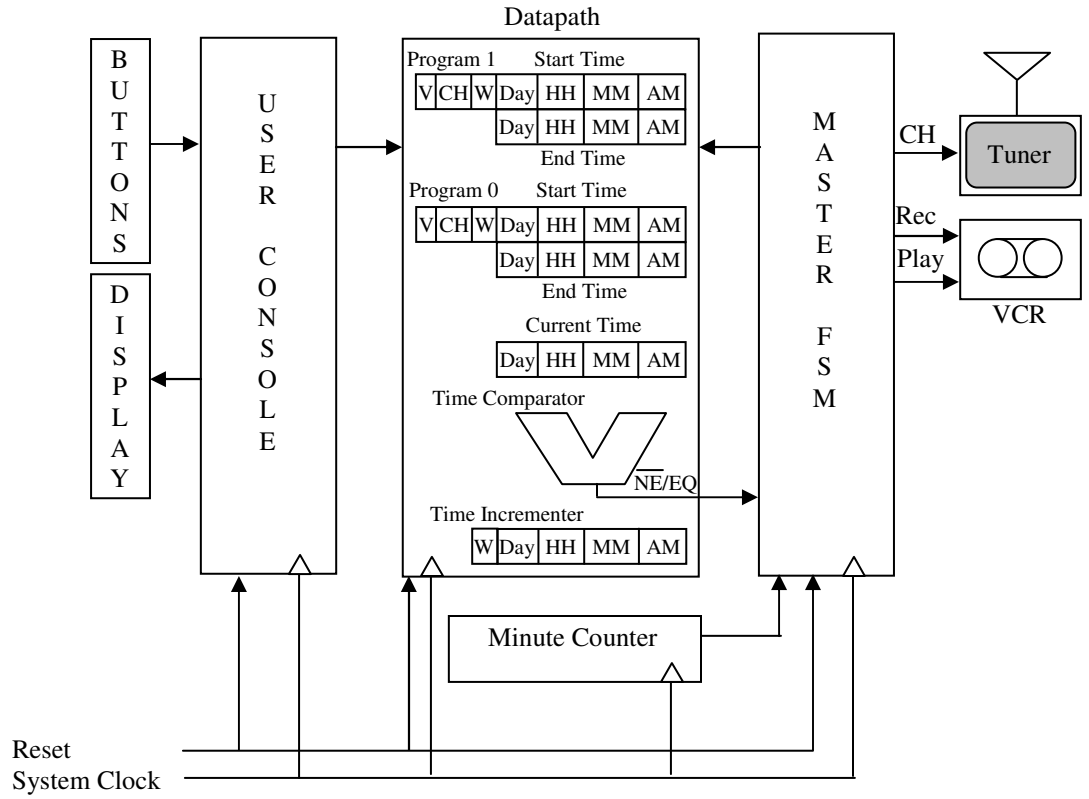
First the Current Time needs to be set. The user pushes the first rocker switch to SET, then places the fourth switch in the "in between" position to indicate that the Current Time is being set, and then rotates the rotary switch to Day. Note that the Program and REC/PLAY rockers are ignored, and that the rotary switch can skip positions as it is rotated. By default, when entering Set mode, the Day is Sunday, the Hour is 12, the Minutes are 00, and morning/evening is PM (also the channel is 99 and week is "this week" for setting programs). The user can skip any part of the time/program specification if the default is what they wanted.

The user presses the Advance button several times until the correct day appears in the Day display. Note that the Advance button will wrap around, e.g., from Friday to Saturday, then back to Sunday. The user then rotates the switch to HH, and repeats the process with the Advance button until the current Hour appears in the Hour display. The next step is to rotate the switch to MM, and set the minutes. If the Advance button is held down rather than pressed for a short time, it will automatically advance. Finally AM/PM is set in the same way when the rotary switch is set to morning/evening (pressing Advance causes AM/PM to toggle between its two possible settings). The Current Time setting is finalized when the fourth rocker is moved out of the "in between" position.

Setting Program 1 to tape Channel 60 on Sunday of next week, at 11:30 AM, works as follows. Place the first rocker switch to Set. Depress the Program rocker switch to indicate Program 1. Press the third rocker switch to StartTime. Place the rotary switch to point to CH. Press and hold down the Advance button until 60 is shown in the channel indicators. Twist the rotary switch to Week. Hit advance once to indicate next week. Advance the rotary switch to Day. Hit advance several times until Su appears in the day indicator. Advance the rotary to Hours. Press advance multiple times until 11 appears. Advance the rotary to MM, and press advance until 30 appears. Advance the rotary to AM/PM and tap the advance button until AM appears.

You would then press the EndTime side of the rocker, repeat the process with the rotary switch/advance button to set the end time (here we would skip the programming of the channel). The program is goes into effect when the first rocker is set to RUN and the second rocker to REC.

The following is a high level block diagram of the VCR Controller. Some details have been left out: this is on purpose, as you will need to supply some of the missing parts!



Program 1/Program 0 are registers that hold the current programs. V indicates whether or not the program stored in that location is valid, CH is two BCD digits (4 bits each) for the Channel, Day is 3 bits for the day of the week, HH is two BCD digits (4 bits each) for the Hour, MM is two BCD digits for the Minutes (4 bits each), and AM is one bit for morning/evening. Both the StartTime and EndTime are stored.

Time Comparator is a combinational logic block that can compare <Day, HH, MM, AM> tuples for equality. Note that HH and MM are 2 x 4-bit BCD digits, while Day is 3 bits and AM/PM is one bit.

Time Incrementer is a combinational logic block that increments by one minute the tuple <W, Day, HH, MM, AM> presented at its input. The updated time is available at its outputs. Note that incrementing the time can cause the week to be decremented!

The System Clock runs at 1 KHz (1 ms period). The *Minute Counter* assets its output for exactly one cycle once every 60,000 cycles of the System Clock.

The outputs from the control are the programming displays on the left, and on the right, the channel to tape to the TV tuner and record and play signals to the video taping mechanism.

For the purposes of this design, you need not worry about handling Play mode correctly.

Question 1. *High Level Description* (20 Points)
To make sure that you understand the function of the VCR controller, write pseudocode that describes its behavior. Declare your variables and use a reasonable and consistent program syntax. Provide separate module descriptions for the Master Controller FSM and the User Console FSM.

Question 2. *Design of Datapath Components* (40 Points)
(i) Design the Minute Counter. Recall that the System Clock is 1 KHz, and that you must assert the Minute Counter output once every 60,000 cycles. You may use hierarchy in your design: show us the gate level implementation of a single counter bit, use this form a building block, and show us your implementation in terms of whatever building blocks you choose. (10 Points)
(ii) Design the DoW/ToD Comparator. This is combinational logic with two DoW/ToD inputs that outputs EQ (1) whenever the two inputs are identical and NE (0) if they differ. Note that some elements of the data to be compared are BCD digits while others are straight binary data. Again, you may use hierarchy in your design: e.g., a BCD digit comparator that can be instantiated and used wherever you need to compare two BCD digits. (10 Points)
(iii) Design the Time Incrementer. It always adds one minute to the <W, Day, HH, MM, AM> presented it as input. It must handle the minute, hour, day, and week wrap arounds correctly. That is, Day wraps from 6 to 0, Hour from 12 to 01, Minutes from 59 to 00, AM/PM toggles when advancing from 11:59 to 12:00, and W is toggled on the transition from Saturday to Sunday. Use hierarchy as appropriate, and draw a logic schematic/block diagram. Please label and describe your inputs and outputs. (20 Points)

Question 3. *Datapath Design* (20 Points)
(i) Complete the design of VCR datapath by showing a block diagram level description with bus-oriented interconnections among the registers and functional units. Be sure to describe in English as well as through logic connections how data moves from the User Console into the Datapath. Also indicate how Reset is handled for each component. (10 Points)
(ii) Tabulate the register transfer operations implied by your datapath design. By this, we are looking for the high level transfers between registers and functional units that can be performed in a single system clock cycle with your datapath design. Provide a short comment with each RT operation to indicate what it is used for. (5 Points)
(iii) Tabulate how each register transfer operation is implemented by lower level microoperations. The latter are the detailed control signals that operate your datapath, such as register Load signals, tri-state enable signals, and operation control inputs to your datapath's functional units. (5 Points)

Question 4. *User Console State Machine* (20 Points)
Develop a state diagram for the VCR's user console defined on Page 3. For simplicity, you may assume anything you need in terms of button behavior, but state these assumptions explicitly in your answer. For example, perhaps you assume that once the Advance button being pressed has been detected by your state machine, you have a control signal to reset this detection. Annotate each state with a description of what it does in terms of register transfer operations inside the datapath.

Question 5. *Master State Machine* (50 Points)
(i) Develop a state diagram for the Master VCR controller. In developing your state diagram, explicitly indicate how it handshakes with the User Console state machine and how it deals with the Minute counter. If you choose to decompose the Master into multiple communicating state machines, tell us your approach. (30 Points)
(ii) Develop a ROM-based encoding for your state machine and a block diagram level implementation of the state machine. This implementation should use as few ROM bits as possible, using encoding techniques on the inputs to reduce the number of address bits and on the outputs to reduce the ROM word width. Briefly describe the rationale of your encoding strategy. (20 Points)