Rensselaer EE EPE CSE
Department Of Electrical, Computer, & Systems Engineering

ECSE 4770 Computer Hardware Design: Lab 3 - Stopwatch Controller

Prepared By: Russell Kraft & Kurt English

Rev A

1. INTRODUCTION

Thus far, laboratory assignments have provided strict implementation guidelines to introduce Verilog & hardware design concepts. This lab introduces a more real-world scenario where a design requirements are given, and students must formulate their own solution using Verilog to code a state machine onto an FPGA. In this lab a controller for a stopwatch timer system will be developed and demonstrated. This requires the following:

- Develop a state machine to respond to inputs and sequence outputs appropriately.
- Exactly match design descriptions, including names and spelling of inputs and outputs.

There will not be a physical demonstration for this lab. Students will thoroughly prove that their design is to specification using simulations ideally through Quartus, but alternatively through IVerilog or Xilinx Vivado. You will create your own test vectors to develop and verify your design. Students will include their final design in a single Verilog file with a top-level module that in name and ports exactly matches those of a pre-defined module so that TAs can compile and verify the student implementation.

2. SCENARIO

You are engineer who must write an FPGA replacement for an industrial stopwatch timer that uses outdated TI 74 series chips. Due to COVID restrictions, you do not have access to the physical equipment, however, a co-worker was allowed to record and share a demonstration video of how the stopwatch is supposed to function. Using this combined with a detailed description of stopwatch requirements, you must implement a Verilog FPGA solution that is a drop-in replacement of the current 74 series system. You decide to begin with the control module for the timing and memory system, with the timers and memory themselves to be over hauled at a later date.

3. STOPWATCH TIMER CONTROLLER REQUIREMENTS

The controller designed in this lab will control an electronic stopwatch. The existing TI 74 series based module supports 15 split times, however your implementation only requires 10 (10 slide switches on the DE10). The stopwatch will be capable of storing up to 10 split times (instantaneous copies of the elapsed time) while maintaining the running time or a split time; and 10 address switches to determine which split is to be saved, viewed, or started.

The following portions of the controller are to be designed for this lab:

1. Stopwatch controller
2. Controller power-up reset circuitry
3. Debouncing circuitry for the START/STOP and COUNTER/SPLIT switches

3.1. Operating Scenario

The operation of the stopwatch timer system is most relatable to that of a runner on track recording lap/split times. The following example indicates the detailed operation of the RC stopwatch:

A runner wishes to find his/her lap times as he/she runs a track. A clock keeper has this lab's stopwatch. The keeper turns on the watch; it automatically resets itself, displaying "00000". The runner starts while the timekeeper presses the START/STOP button and views the running time. At the end of the first lap, the keeper hits ADDRESS SWITCH #1 which causes the runner's first lap time to be saved in memory location 1. At the end of the second lap, the keeper hits ADDRESS SWITCH #2 and saves the runner's total time for the first and second laps ("the runners split timed") in memory location 2. The runner stops at the end of the fourth lap and the timekeeper hits the START/STOP switch to stop the clock.

The runner now wishes to see the time for the first lap, so the keeper presses the COUNTER/SPLIT switch, and then presses the address switch of the split the runner wishes to see: ADDRESS SWITCH #1. Then to show the runner the elapsed time after lap 2, the keeper presses ADDRESS SWITCH #2.

The runner wants to run on the track again, but he wants to start the stopwatch with the time he achieved for three laps. The keeper presses ADDRESS SWITCH #3, which was used to store the cumulative elapsed time at lap three, to get the split on the LED display. Now, when the runner starts the keeper will hit the START/STOP switch again and the clock will continue counting from the third lap split.

3.2. Stopwatch Function Summary

1. Power on: when power is applied, the controller should start in a known state and the counter display on the stopwatch module should read zero.
2. To start from zero:
    a. When the power is turned on and the START/STOP switch is pressed, the counter will start counting from zero.
    b. If the counters are stopped and the COUNTER/SPLIT switch has not been pressed to view any splits, the counter will be started from zero by pressing the START/STOP switch.
    c. If the counters are stopped and the COUNTER/SPLIT switch has been pressed to view splits, then the counter will start from zero when the COUNTER/SPLIT switch is pressed again followed by the depression of the START/STOP switch.
3. To stop the counter: the START/STOP switch is depressed while the counter is running.
4. To store a split time: press an ADDRESS SWITCH while the stopwatch is running. The split time will be stored in the memory location indicated by the ADDRESS SWITCH.
5. To recall a split time: (This can only be done while the counters are stopped.)
    a. Press the COUNTER/SPLIT switch and then the ADDRESS SWITCH corresponding to the desired split time.
    b. If the COUNTER/SPLIT switch was pressed to view a split previously, then only the ADDRESS SWITCH need be depressed to view any subsequent split time.
    c. If the COUNTER/SPLIT switch is pressed again the stopwatch should return to the "display counter" mode with the final stopped time displayed on the LED display.
6. To start from split: Follow the procedure previously defined to recall a split. When the desired start split is on the stopwatch LED display, the START/STOP switch should be pressed and the stopwatch should start from the displayed split.

3.3. Description of Stopwatch Hardware

A basic block diagram of the stopwatch system is shown in Figure 1. As shown in figure 1, the stopwatch is made up of two sections.

- External timing and memory hardware, which will be created and modified in lab 4.
- Stopwatch control module, which will be created in this lab.
- The separation of these two parts is clearly delineated in the figure.

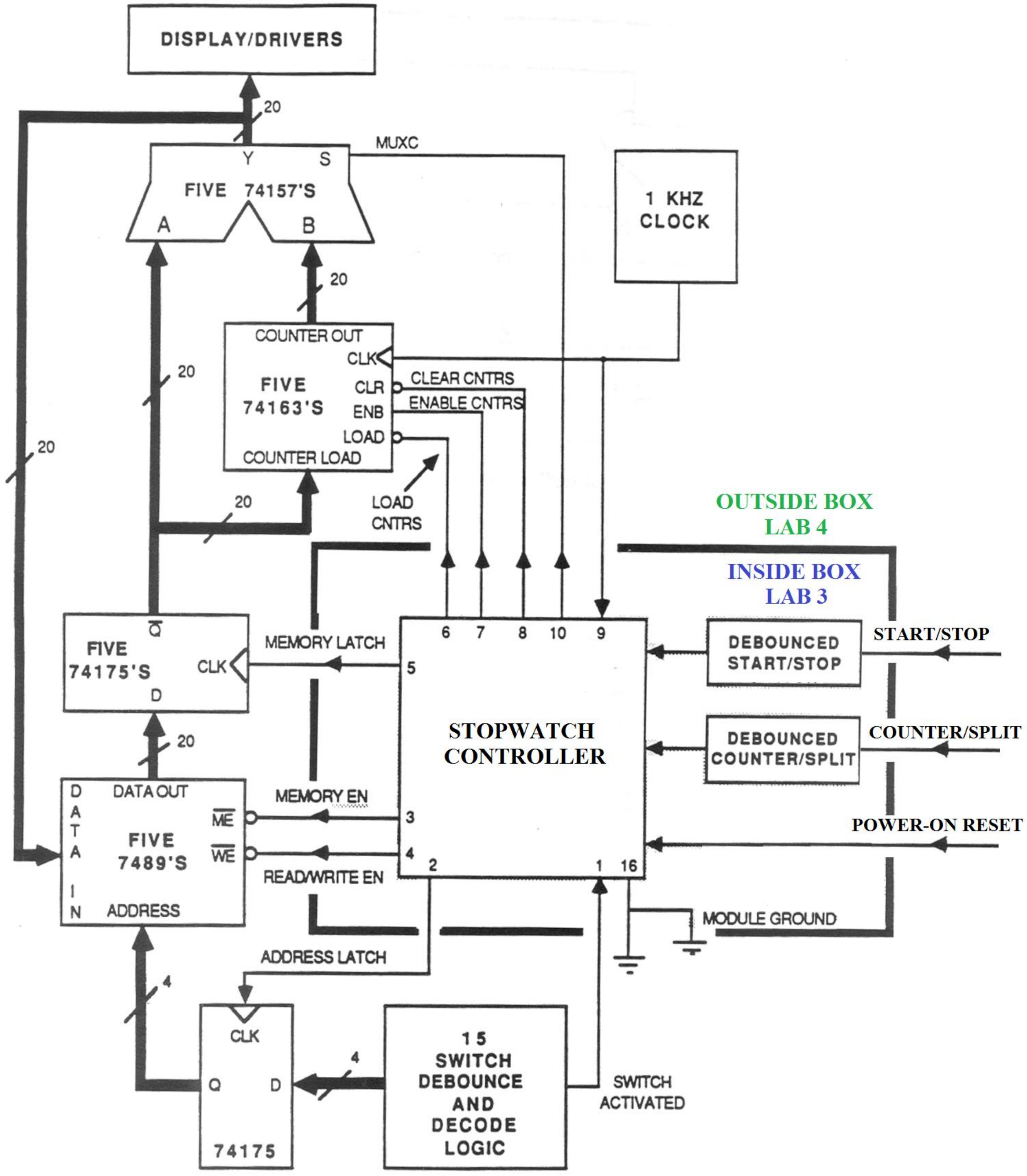


Figure 1: Stopwatch block diagram

3.4. Stopwatch Module Subsystems

Outside of the controller, the stopwatch module is composed of six major subsystems:

1. Display and drivers
2. Multiplexer
3. Counters
4. Split memory
5. Address switch logic
6. 1 kHz clock

1. The DISPLAY/DRIVERS block provides the necessary logic to decode BCD and drive a seven segment LED display.
2. The MULTIPLEXER block consists of five 74157 chips. This block selects the data that is applied to the LED display , which also leads to the memory input.
3. The COUNTERS block consists of five 74163 counter chips and some additional reset logic to provide the minutes, seconds/tenths/hundredths of a second format. All the counters work synchronously with the 1 kHz clock.
4. The SPLIT MEMORY block has two parts. The first, Main Memory, uses five 7489 memory chips while the second is memory latch consisting of five 74175 chips. The main memory holds the splits from the COUNTER block in the locations specified by t he ADDRESS SWITCHES. The memory latch has two functions: it inverts the logic levels present in the main memory and it holds the last recalled split regardless of the output from the main memory.
5. The ADDRESS SWITCH logic provides the switch debouncing and encoding plus an address latch that holds the address generated by the ADDRESS SWITCHES. The timing of the ADDRESS SWITCH logic is shown in figure 2.
6. The 1 kHz clock is a 555 timer running at 2 kHz with a 7474 flip-flop acting as a divide-by-two stage. The 7474 divide-by-two circuit provides a symmetrical square wave clock of 1 kHz.

3.5. Stopwatch Control Module Interface

Currently, a ribbon cable with a 16-pin DIP connector is attached to the stopwatch module. This cable carries all the interface signals needed to control the stopwatch module. The interface signals are listed in Table 1.

Table 1: Summary of Stopwatch Module Interface Signals

Pin #	Name	Mnemonic	Type	Loading
1	Switch Activated	SA	10 ms pos. pulse	N/a
2	Address Latch	ADDLATCH	Rising edge triggered	1
3	Memory Enable	ME	Active low	5
4	Read/Write	R/W	Write low	5
5	Memory Latch	MEMLATCH	Rising edge triggered	5
6	Counter Load	LD	Active low	5
7	Counter Enable	E	Active high	5
8	Counter Clear	CLR	Active low	5
9	1 kHz Clock	CLK	1 kHz square wave	N/a
10	Multiplexer Select	MUXC	Counter High	5
11-15	Unused	N/a	N/a	N/a
16	Ground	GND	N/a	N/a

3.5.1. Verilog Requirements

For this lab, you will not be connecting to existing hardware. You will be verifying all aspects of design through simulation . You will create your own test vectors to develop and verify your design. Additionally, your Verilog will be compiled and simulated by TAs to ensure that all functionality is present. To do this, you will do the following:

- You will have all Verilog located in one file
  - At the top of the file you will include you and your lab partner's name in comments
  - This file will be named **stopwatch_controller.v**
- Input and Output requirements
  - No inputs or outputs will be registered
  - You will have the following inputs and outputs with names exactly matching those of the following table:

Table 2: Stopwatch Controller & Verilog Names

Signal Name	Verilog Name	Type
Start/Stop	SS	Input
Counter/Split	CS	Input
Power-On Reset	RST	Input
Switch Activated	SA	Input
1 kHz Clock	CLK	Input
Address Latch	ADDLATCH	Output
Memory Enable	ME	Output
Read/Write	MW	Output
Memory Latch	MEMLATCH	Output
Counter Load	LD	Output
Counter Enable	E	Output
Counter Clear	CLR	Output
Multiplexer Select	MUXC	Output

3.5.2.Switch Activated (SA)

SA provides a 10-ms positive pulse when one of the split memory address switches has been pressed. While SA is high, a valid address is presented to the address latch. The address switch timing is show in Figure 1.

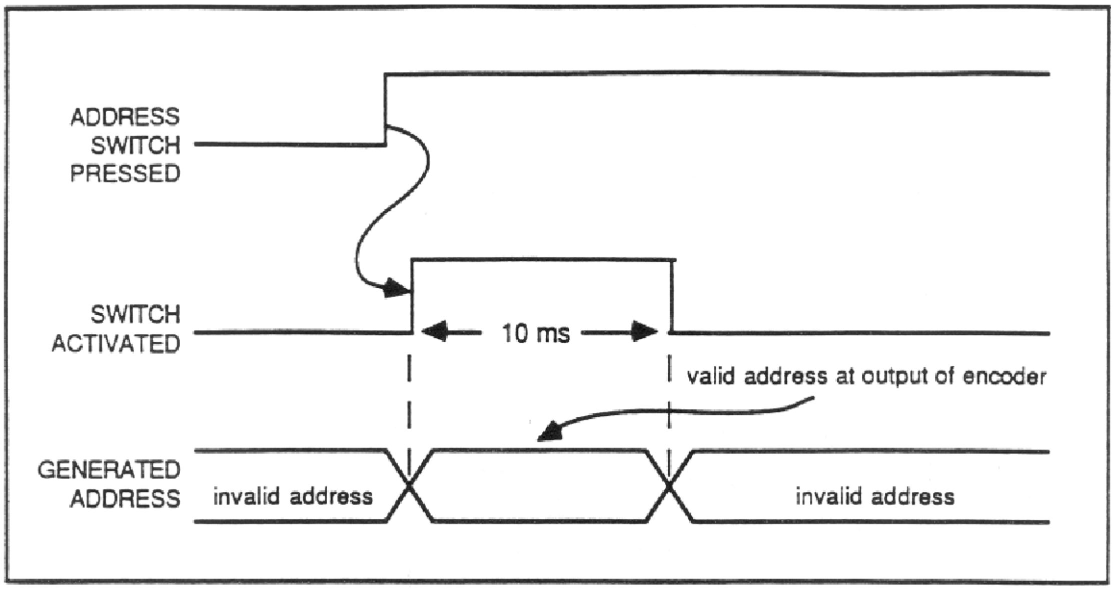


Figure 2: Address Switch Timing

- ADDLATCH is a positive edge triggered signal that latches the data presented by the address switch matrix.
- ME is an active low signal that enables the both the input and output ports of the split ti me memory.
- RW is a signal which enables either the input or output port of the split time memory. When RW is high, the output is enabled and data can be read. When low, the input is enabled and data is written.
- MEMLATCH is a rising edge triggered signal that latches the data read from the split-time memory.
- LD is an active low signal the loads the counters with the output of the memory latch. The load is synchronous with the clock.
- E enables the counters for incrementing with the clock. When low, the counters retain their last value.
- CLR clears the counters, setting the output to zero. The clear is synchronous with the clock.
- CLK is the 1 kHz clock for the entire circuit. The counters actually have a hidden 6th digit driven by this clock that has no display. This acts as a divide-by-10 circuit to drive the hundredths digit of the display.
- When the MUXC signal is high, the multiplexers select the counter output; when low the memory output is selected.
- GND is connected to ground on the stopwatch module. This is to ensure that the interface signals are in reference to a common ground.

3.5.3. Start/Stop and Counter/Split Debouncing

Start/Stop and Counter/Split are provided through active high switches. These switches in practice will be held high for much longer than the 1 kHz clock period. You must prevent this from causing a race between FSM states with only one single input being detected. This can be done by either of the following means:

- Include additional wait states which wait for the input to drop low prior to proceeding.
- Debounce logic to clamp the input to be no longer than one clock period.
- Have SS and CS be positive edge triggered. This complicates timing with the CLK for the FSM.

4. DEMONSTRATION VIDEO

A video can be found demonstrating stopwatch controller operation on the existing TI 74-series stopwatch module can be found on LMS at the following link:

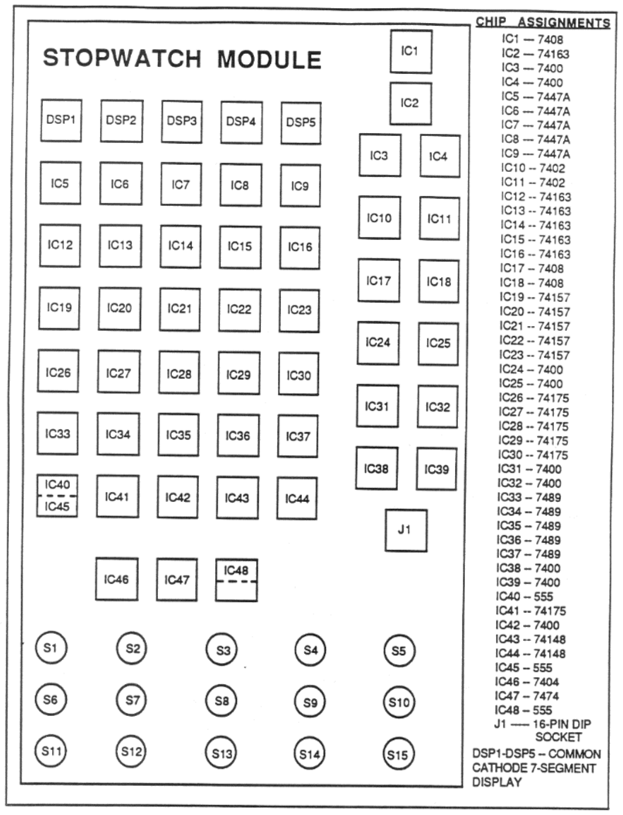
https://lms.rpi.edu/webapps/blackboard/execute/content/file?cmd=view&mode=designer&content_id=_209574_1&course_id=_7160_1&framesetWrapped=true

5. SUMMARY & CONCLUSIONS

You must submit both your Verilog code and report on LMS. Your report is to contain the following material:

- Names
  - Both you and your lab partner if you have one
- Preface/Abstract
  - Describe lab requirements
- Overall design description
  - State the overall structure of your lab submission and how this meets requirements
- State Diagram
  - Include a full state diagram of the state machine you implemented to complete this lab assignment
- Schematics
  - You must submit a meaningful schematic, or multiple schematics of your work if necessary so that the overall structure of your design is apparent
  - These can be generated through the RTL Viewer in Quartus or by automated means in other tools
  - Hand drawn schematics are also acceptable
- All Verilog & supplemental files
  - Explain how your code implements all required features
  - Be detailed. Include snapshots of your code and describe its function
    - Several paragraphs of text are expected in total
    - Use graphics to aid in presentation
- Simulated timing diagrams
  - Describe how each major requirement is demonstrated in the timing diagrams
    - This likely will require multiple diagrams
  - Demonstrate how the timing on each output is correct
    - State in words why what is observed in simulation is correct
- Verification
  - State how your implementation was verified
- Conclusion

6. ADDITIONAL MATERIALS



CHIP ASSIGNMENTS

IC1	7408
IC2	74163
IC3	7400
IC4	7400
IC5	7447A
IC6	7447A
IC7	7447A
IC8	7447A
IC9	7447A
IC10	7402
IC11	7402
IC12	74163
IC13	74163
IC14	74163
IC15	74163
IC16	74163
IC17	7408
IC18	7408
IC19	74157
IC20	74157
IC21	74157
IC22	74157
IC23	74157
IC24	7400
IC25	7400
IC26	74175
IC27	74175
IC28	74175
IC29	74175
IC30	74175
IC31	7400
IC32	7400
IC33	7489
IC34	7489
IC35	7489
IC36	7489
IC37	7489
IC38	7400
IC39	7400
IC40	555
IC41	74175
IC42	7400
IC43	74148
IC44	74148
IC45	555
IC46	7404
IC47	7474
IC48	555
J1	16-PIN DIP SOCKET

DSP1-DSP5 -- COMMON CATHODE 7-SEGMENT DISPLAY

Figure 3: Board Chip Layout

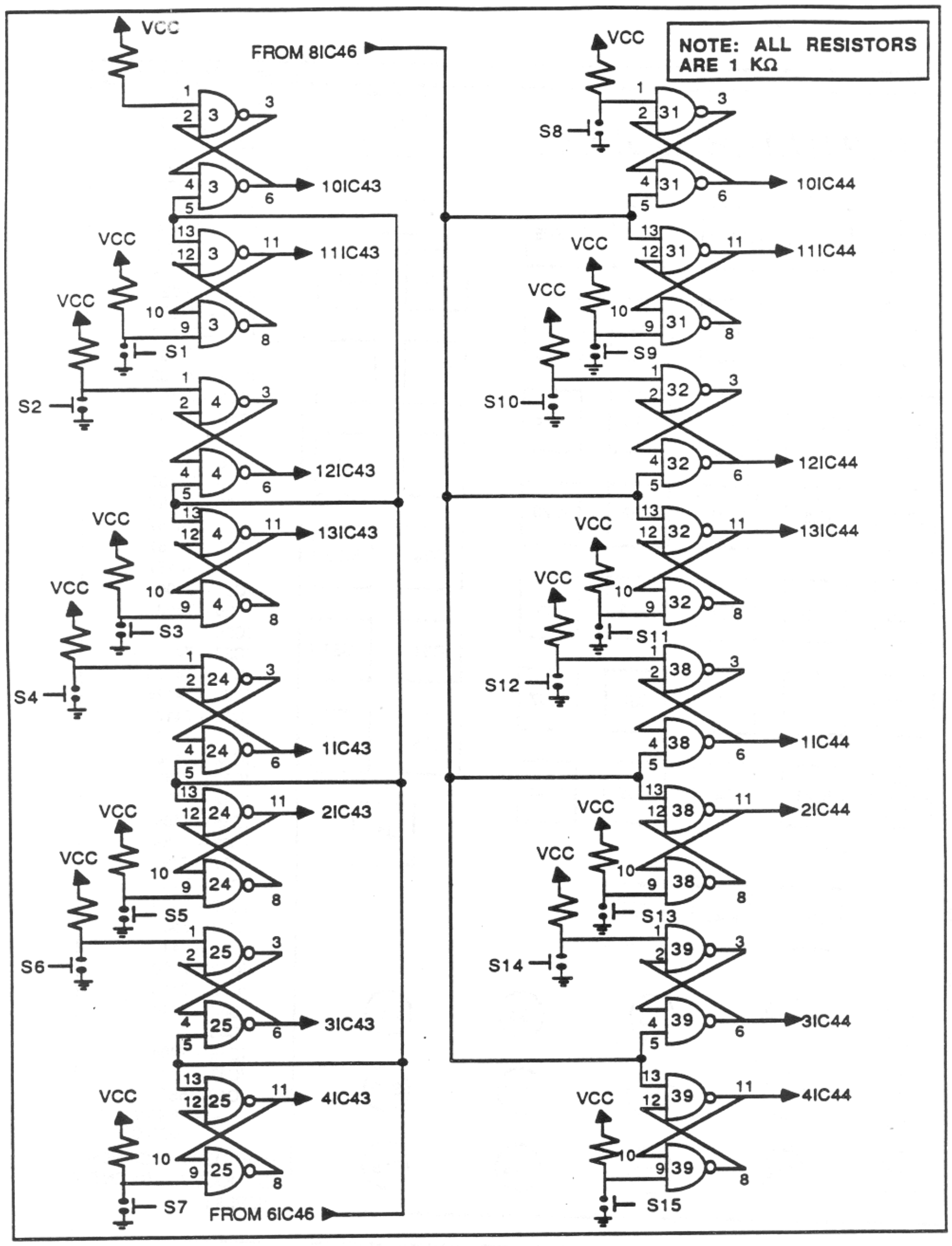


Figure 4: Address Switch Debouncing Logic

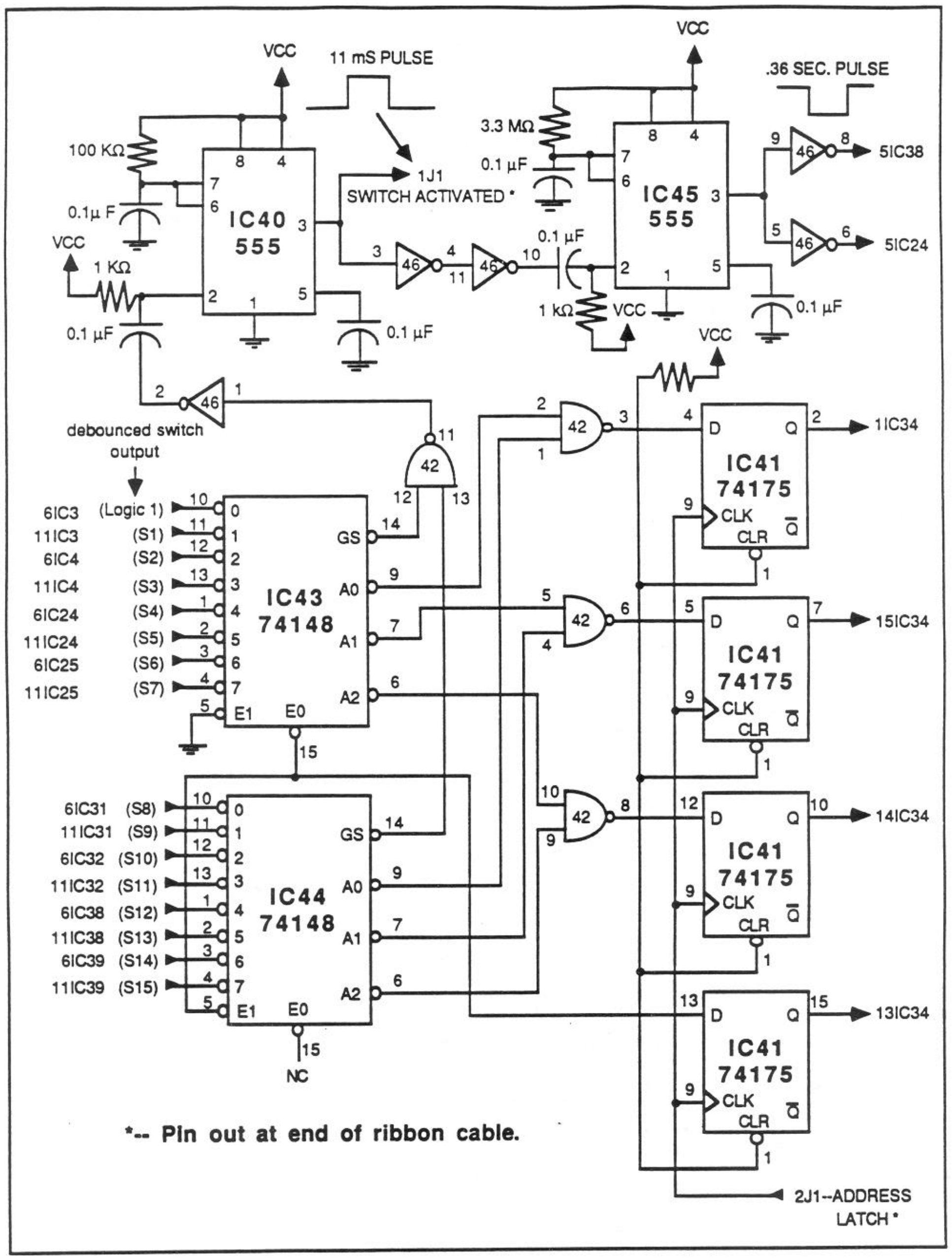


Figure 5: Encoding logic, SA Signal Generation, And Address Latch

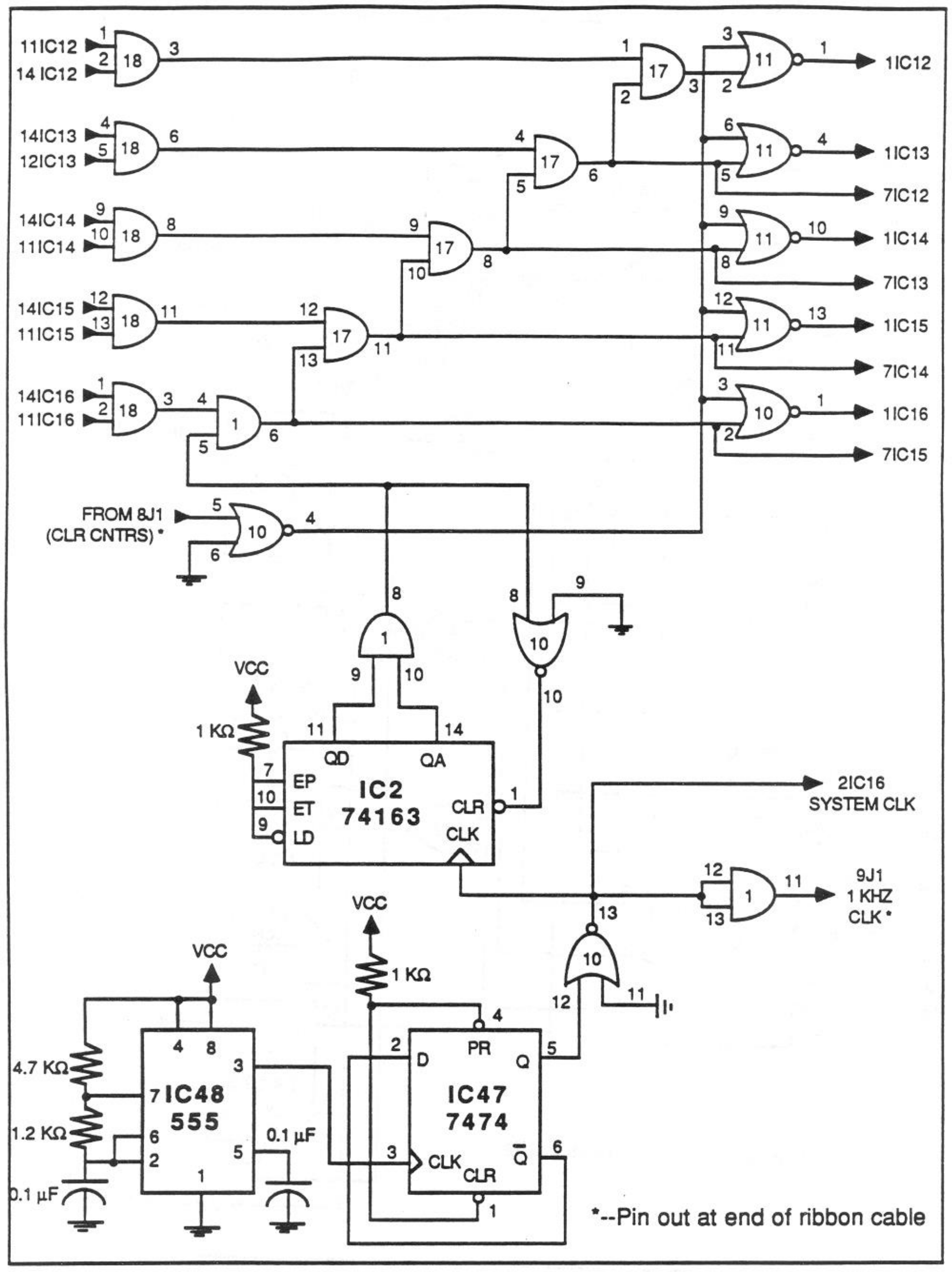


Figure 6: Counter Enabling Logic and 1 kHz Clock Generator

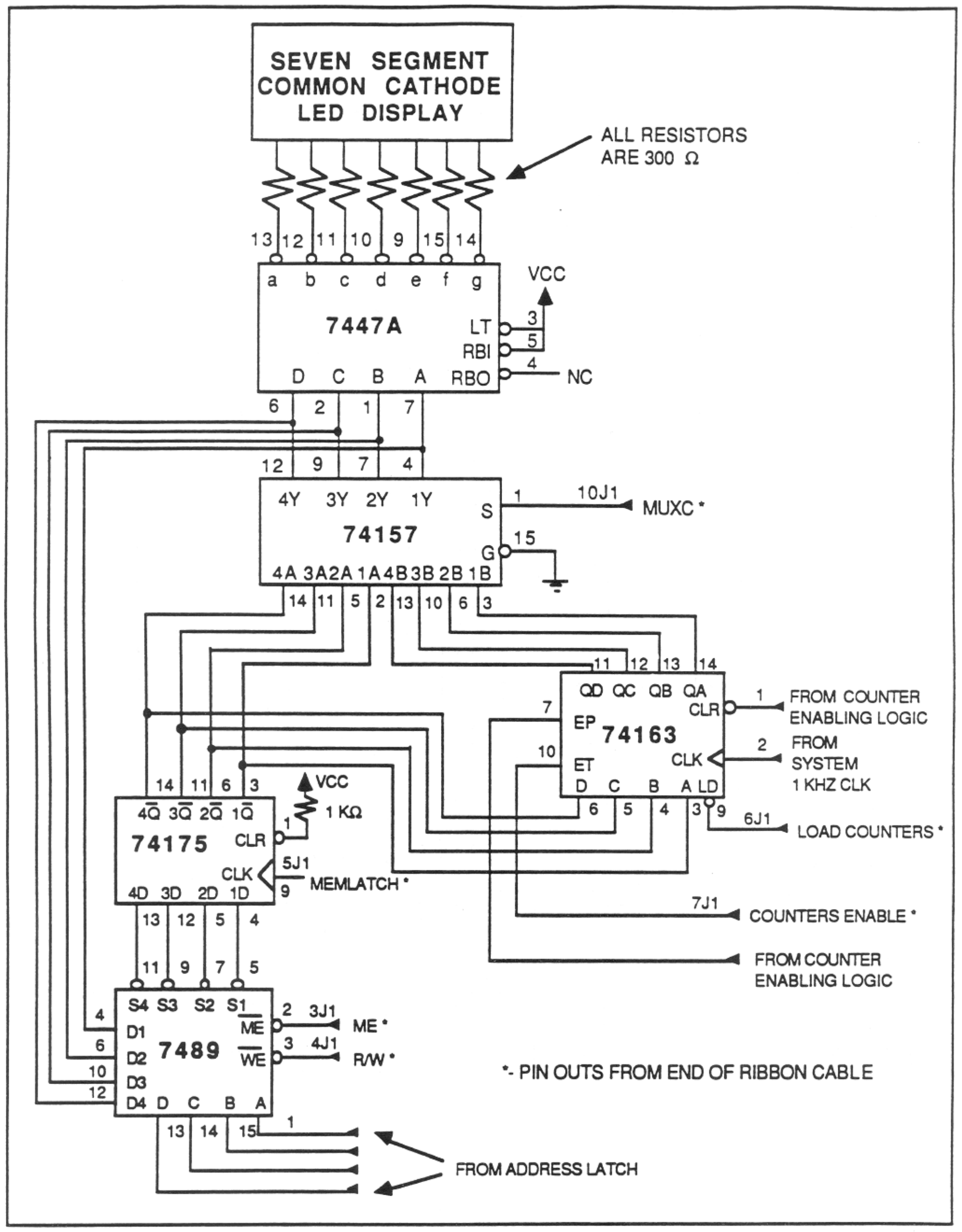


Figure 7: One Digit of Stopwatch Module