Jason Fong
William Durley
4-11-01

VLSI Final Project: Alarm Clock

Functional Overview

The goal of this project was the design of a 24-hour alarm clock. Our chip consists of two major sections. The first of these is the register file, which stores and updates the time. The other section consists of a seven-segment display output decoder, a comparator, and the input signal processing circuits.

The basic functionality of the chip consists of storing, updating, and displaying the hours and minutes of the current time, as well as allowing the user to change these values at will. In addition to this basic functionality, the chip stores a second set of hour and minute values, to be used as an alarm. The chip takes as input two non-overlapping clocks and the user input signals, and provides as output the necessary signals to drive four digits of a seven-segment display, as well as one bit to signal that the alarm should activate. A summary of input and output signals is provided below.

Signal Name	Type	Function
Phi_1, Phi_2	Input	2 non-overlapping clock phases
Hour, Min, Select	Input	User input (sets the time/alarm time)
AlarmStop	Input	Deactivates Alarm
SegA_H – SegG_H	Output	Display output for hour
SegA_M – SegG_M	Output	Display output for miute
Vdd, Gnd	Power/Ground	Power/Ground

Table 1: summary of input and output pins

The register file section of the chip contains six base-ten, synchronous counters, designed to reset at 60. These counters store the hour and minute for the alarm, and the hour, minute, second, and 1/60 of a second for the clock. It is necessary to keep track of

the time in increments smaller than seconds in order to facilitate user input more often than once per second. The time data is stored and updated in base ten rather than base two. This is accomplished via the use of one four-bit register for each digit of a base ten number. Storing the time in base ten is preferable, as it makes output decoding much simpler. The register file takes the two phase clock and preprocessed inputs as input, and outputs the hours and minutes of the time and alarm.

The rest of the chip is predominantly occupied by the output decoder, which takes the base 10 time values from the register file and converts them into seven segment display outputs. The input processing unit contains a pulser, which takes the user input signals and sends a one-cycle long pulse to the register file each time a user input turns on. Without the pulser in place, any attempt on the part of the user at resetting the time would result in the hours or minutes incrementing 60 times per second, rendering the inputs useless. Table 2 summarizes the area and design time spent on each facet of the chip, as well as design verification status of each facet.

Signal Name	Design Time (Hours)	Area (λ^2)	DRC Pass	ERC Pass	NCC Pass	Simulates Correctly
AND4	1	14356	Yes	Yes	Yes	Yes
Bitslice	3	481632	Yes	Yes	Yes	Yes
AlarmHold	4	78746.875	Yes	Yes	Yes	Yes
And4_2	3	7905	Yes	Yes	Yes	Yes
CarryProc	2	10230	Yes	Yes	Yes	Yes
Counter	7	35200	Yes	Yes	Yes	Yes
Counter10	5	257291	Yes	Yes	Yes	Yes
Counter6	5	209991.5	Yes	Yes	Yes	Yes
Compare	2	46511.5	Yes	Yes	Yes	Yes
Dispdecode	8	423384	Yes	Yes	Yes	Yes
iprocess	7	204877	Yes	Yes	Yes	Yes
Mux2	0	5141	Yes	Yes	Yes	Yes
or9	4	21388.5	Yes	Yes	Yes	Yes
pulser	8	40303.5	Yes	Yes	Yes	Yes
Regfile	40	3383799	Yes	Yes	Yes	No
TopLevel	2		No	No	No	No
XOR	2	7029	Yes	Yes	Yes	Yes
Xor_jf	1	8972.5	Yes	Yes	Yes	Yes

Table 2: summary of facet data

Simulation Results

In order to verify the functionality of the chip, simulations were performed on the two sections individually. Unfortunately, we were unable to get the register file layout simulating properly, so top-level simulation of the entire chip was not possible. The register file's schematic simulated perfectly, however, and the layout passed the network consistency check, so we will provide simulation results of the register file schematic.

In order to test the register file, a set of test cases was applied. First, file was reset, and then allowed to run as the second and minute outputs were monitored. The schematic performed perfectly for this test, and the result can be seen in figure one. After this preliminary test, we tested cases that were likely to fail. All rollovers from one counter to the next were tested, and finally, the large rollover from 23:59 to 00:00 was tested. All tests were passed, and a selection of the resulting test waveforms is provided below. At the top of the list of waveforms are six busses, each containing one digit of the second, minute, or hour. For example, the bus sec_hi stores the tens digit of the seconds. Clocks and input signals are shown below the outputs.

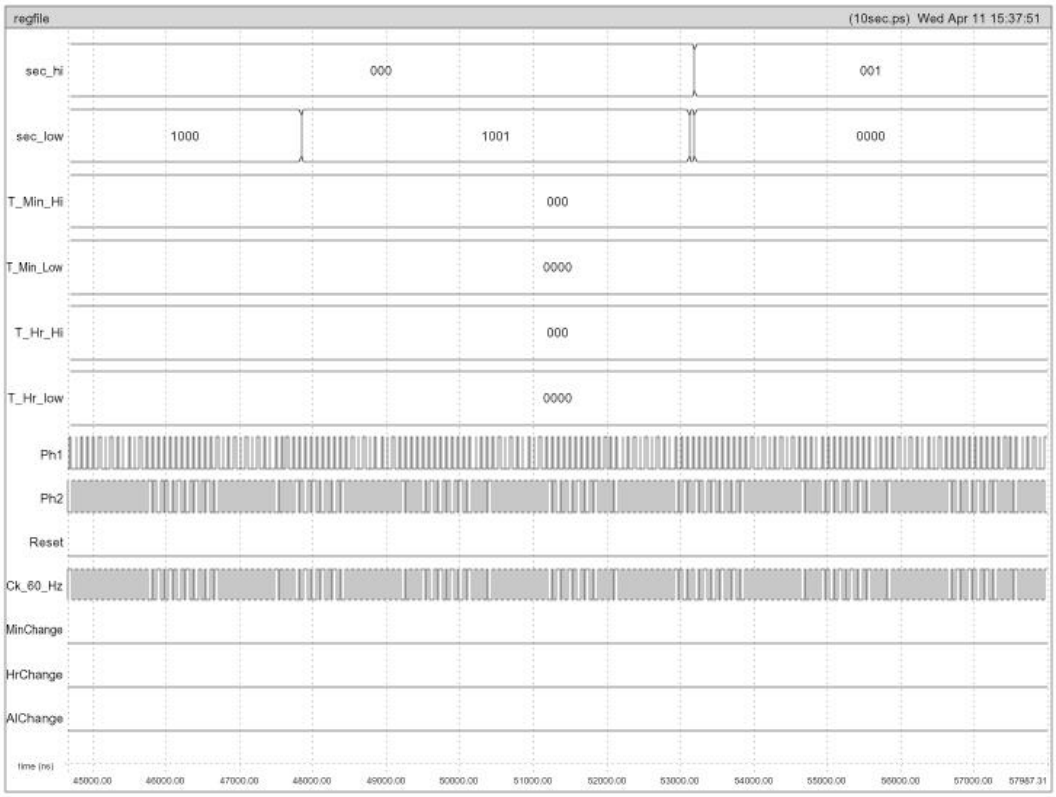


Figure 1: Operation of second counter with no input. Demonstrates that the seconds are incremented once per 60 cycles, as desired

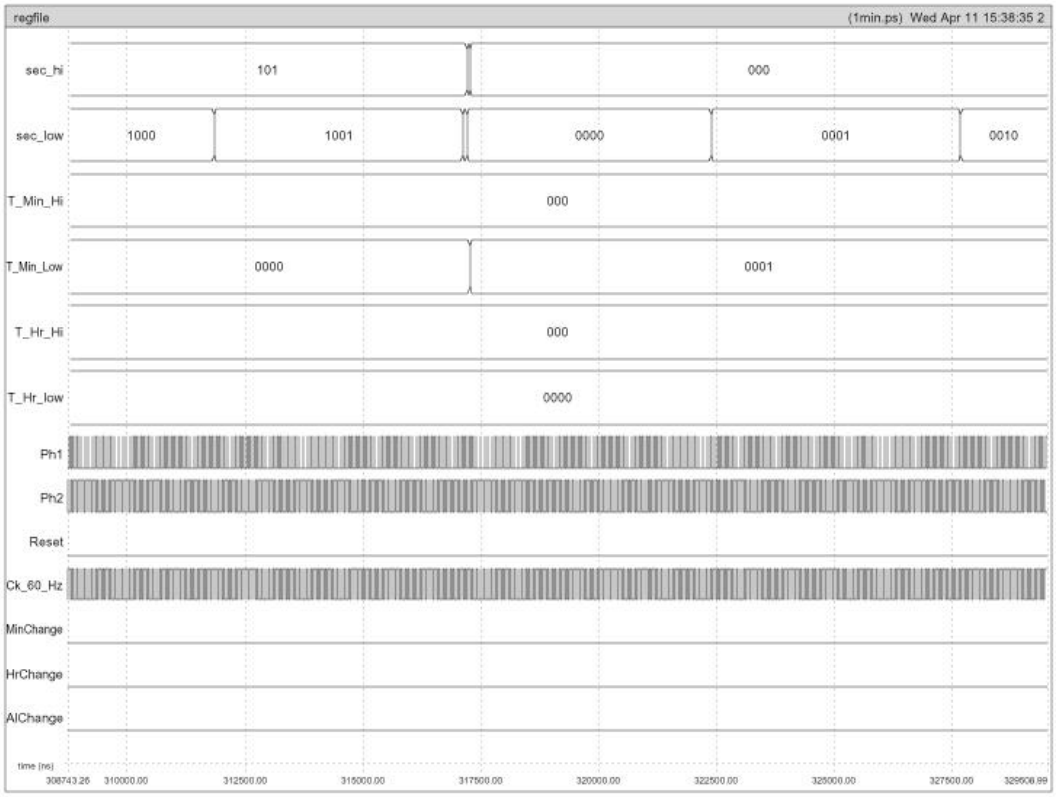


Figure 2: Second to Minute Rollover. Demonstrates resetting of seconds at 60, as well as accumulation of one minute.

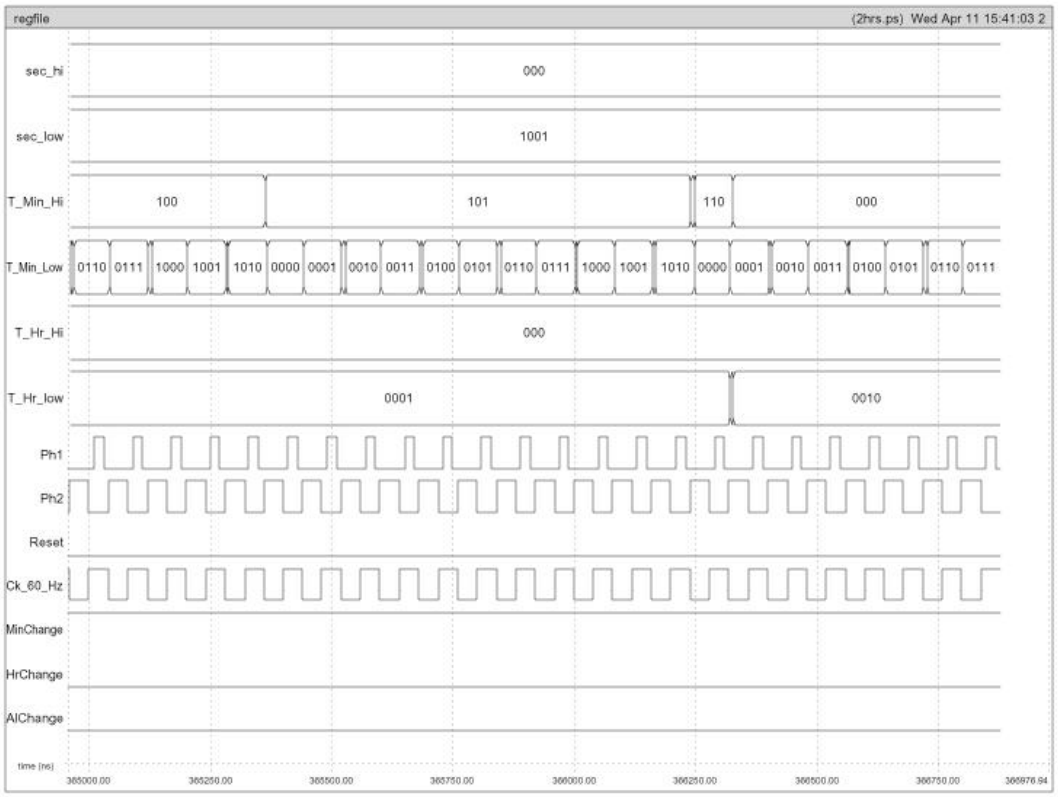


Figure 3: Minute input. Demonstrates the functionality of the minute change input. Also demonstrates rollover from minutes to hours at accumulation of 60 minutes.

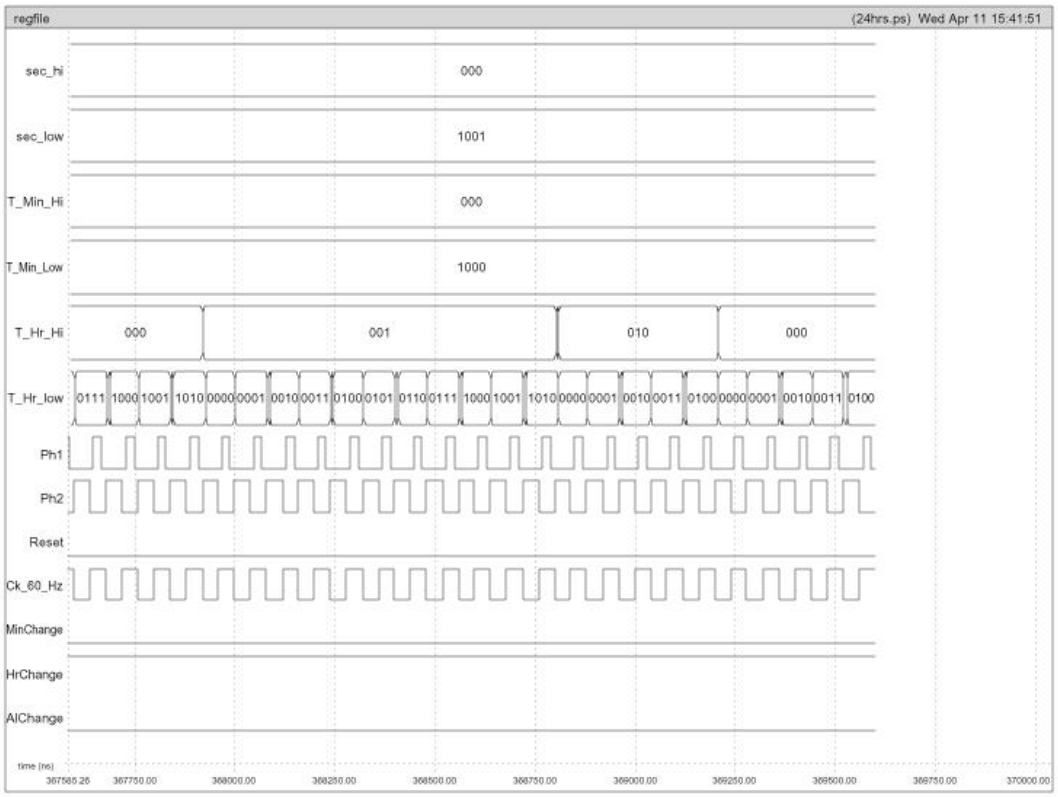


Figure 4: Hour input. Demonstrates the functionality of the hour change input and the reset of the hour at accumulation of 24 hours.

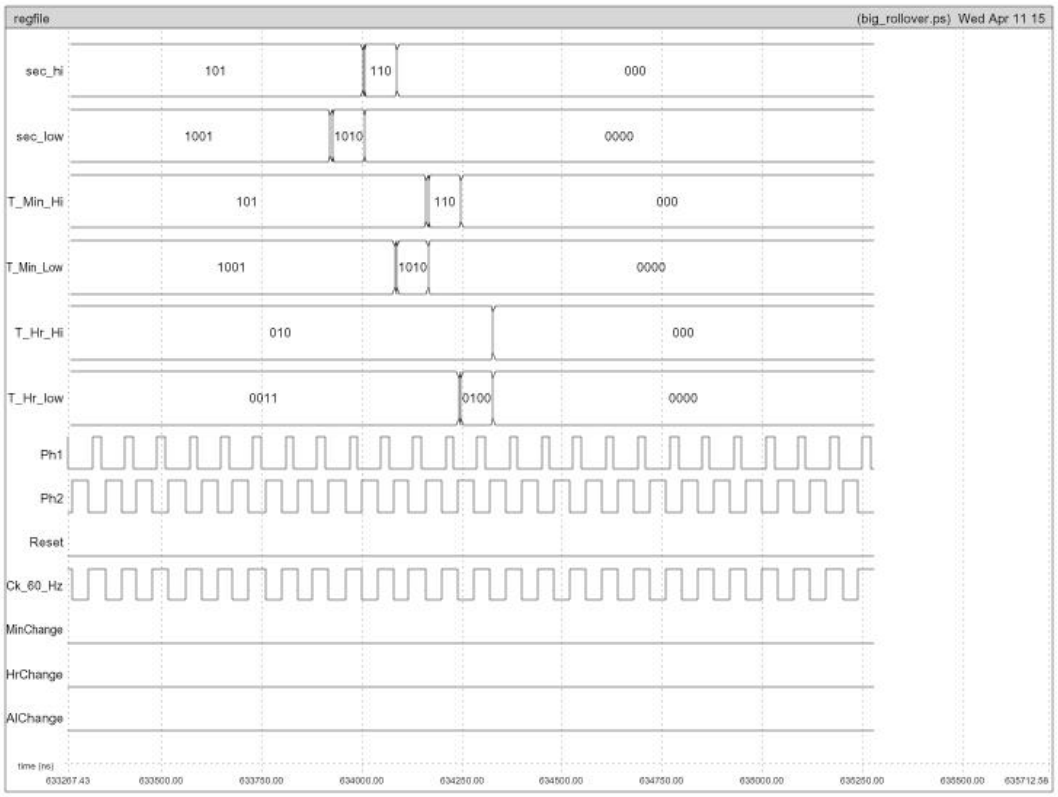
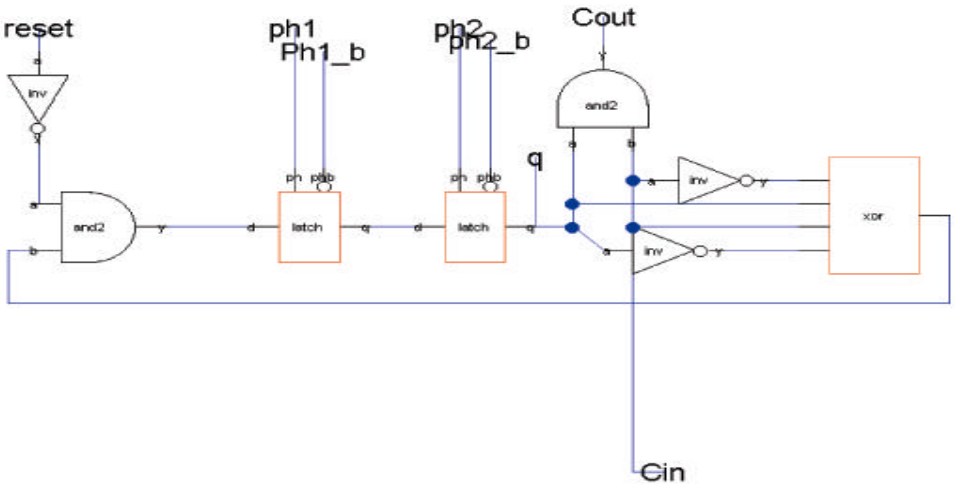


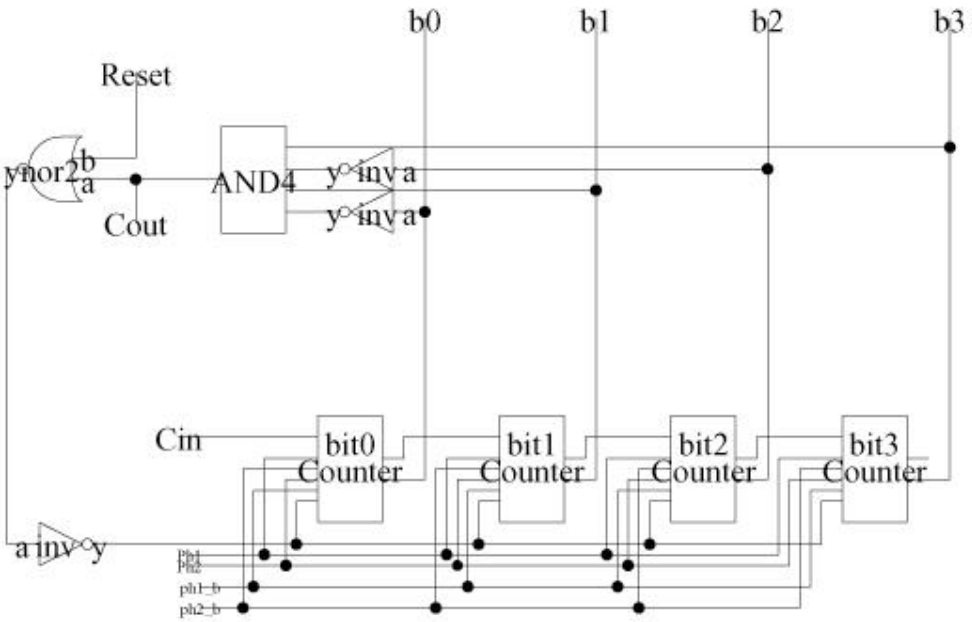
Figure 5: 24-hour rollover. Demonstrates the rollover from 23:59:59 to 00:00:00

Schematics

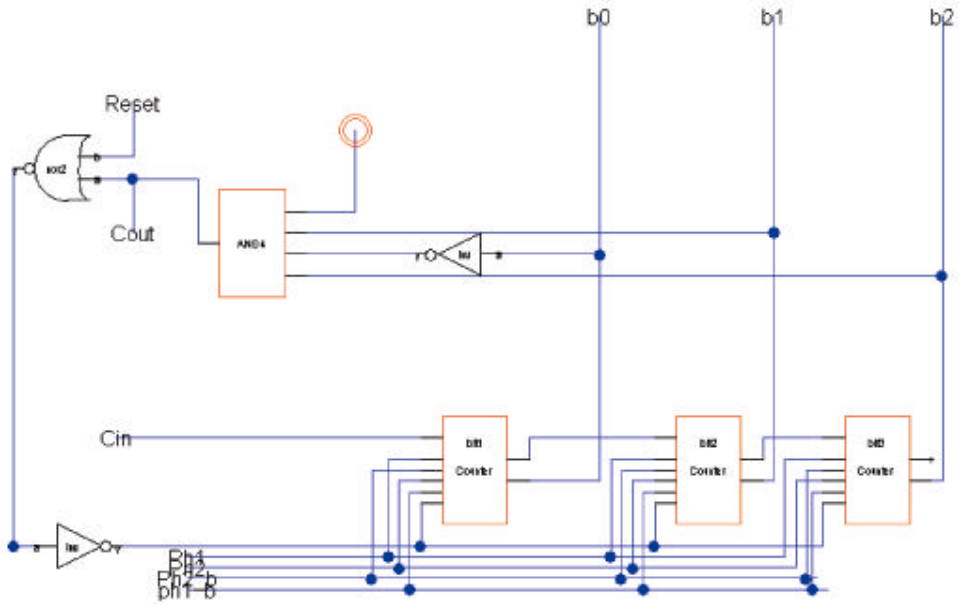
Below are the schematics for all of the major facets of our chip. Very simple schematics, consisting of only a few low-level logic gates, have been omitted.



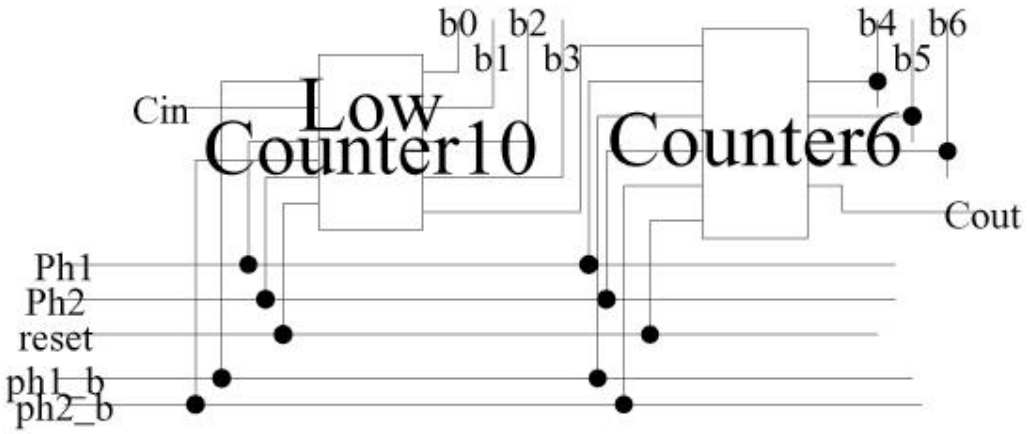
Counter: The basic one-bit synchronous counter used in the register file



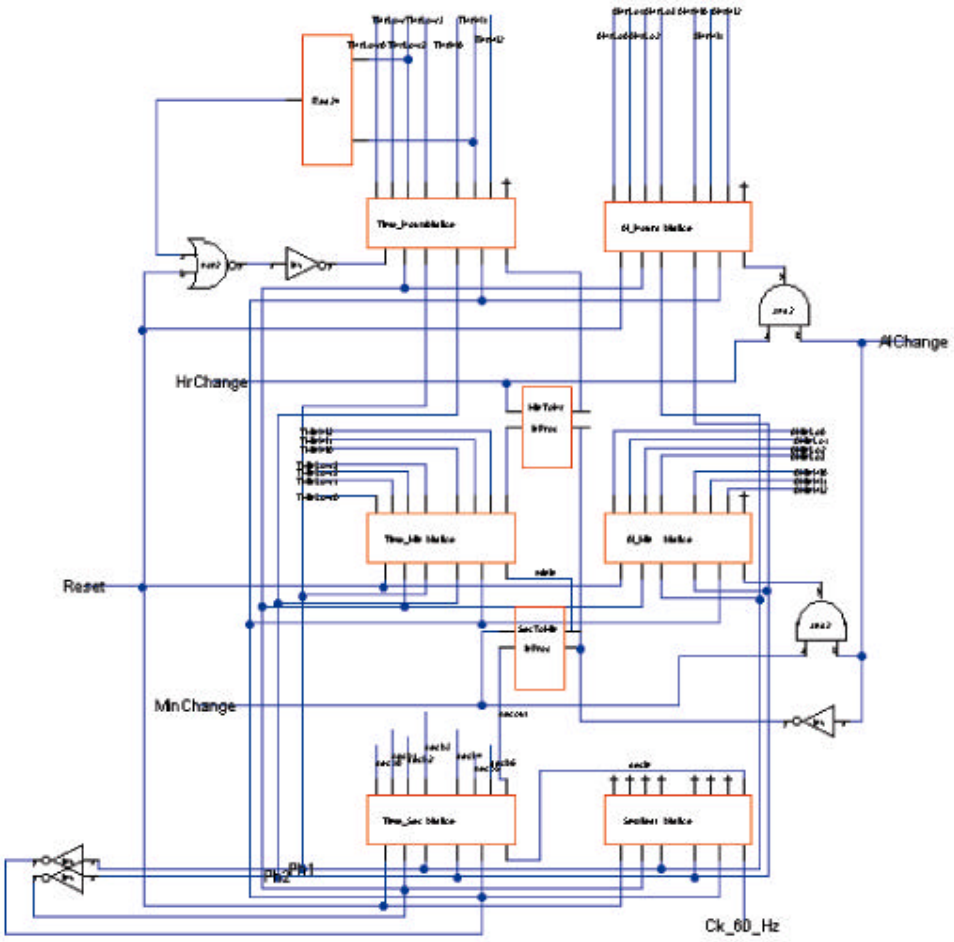
Counter10: A four bit synchronous counter that resets and asserts carry out at a value of 10.



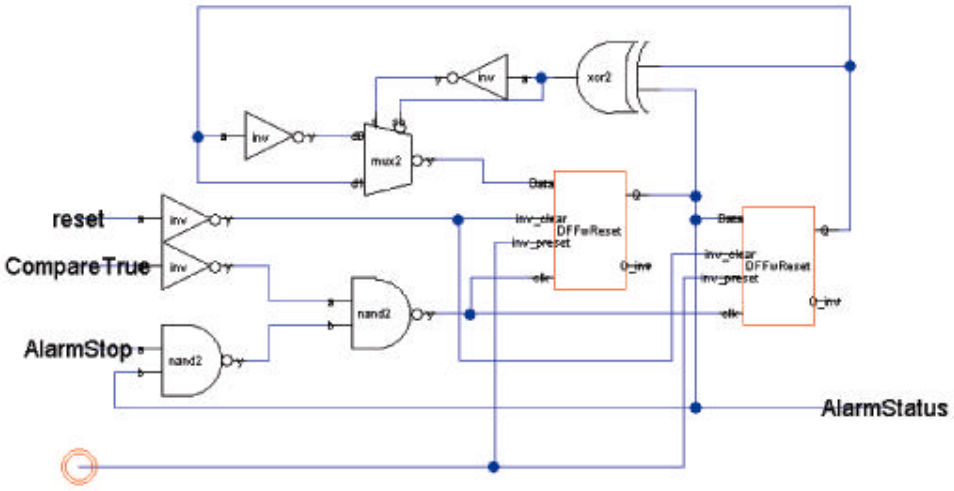
Counter6: A 3 bit counter that resets and asserts Cout on 6



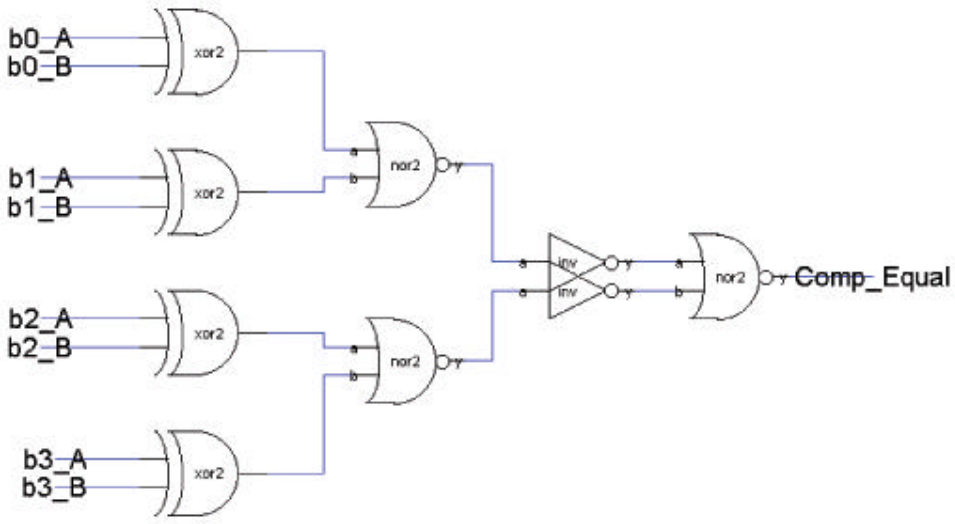
Bitslice: a 7-bit, base 10 counter that resets and asserts Cout at a value of 60.



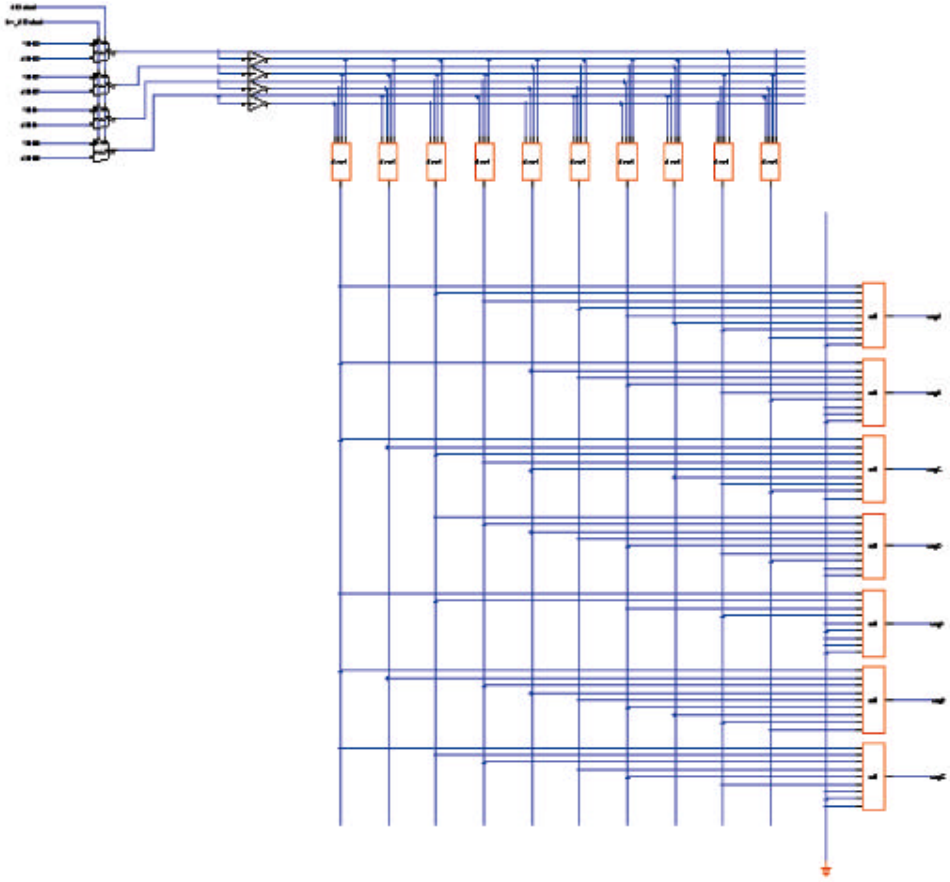
Regfile: The register file. Stores and updates the current hours, minutes, and seconds, and stores the alarm time.



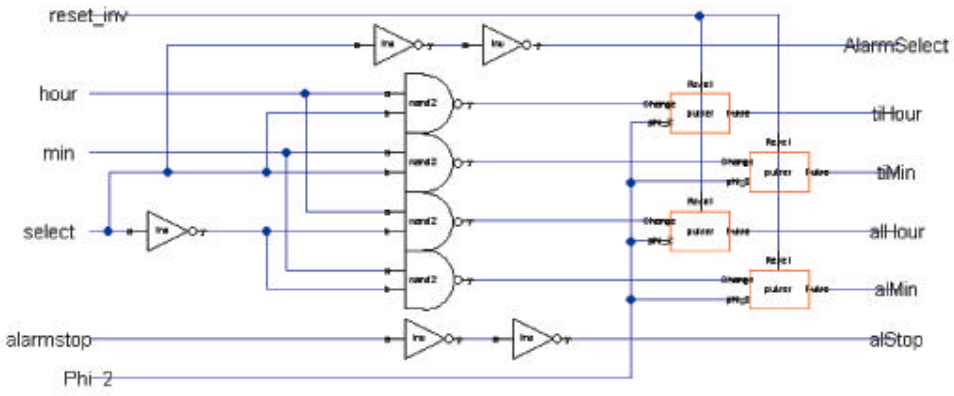
Alarm Hold: when CompareTrue is high, holds the alarm on until AlarmStop is asserted.



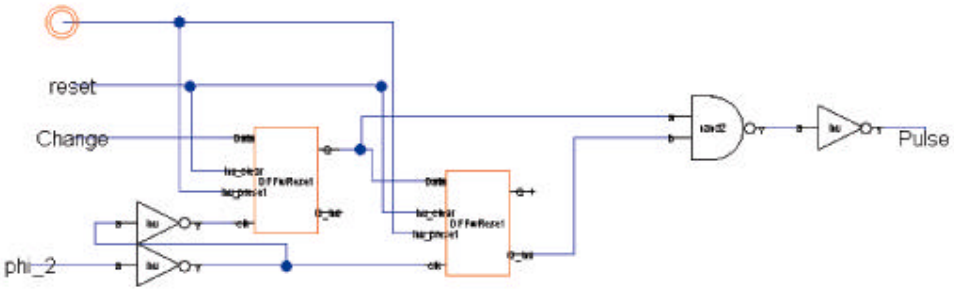
Compare: Compares two 4-bit numbers, and pulls output high if they are the same.



Decoder: Decodes Time for output to 7-segment display

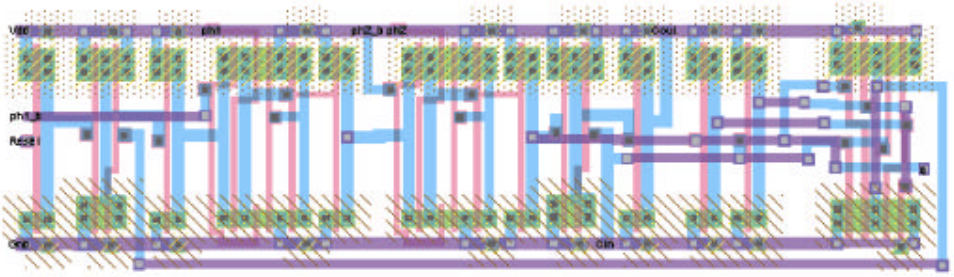


Iprocess: prepares input to be sent to the register file

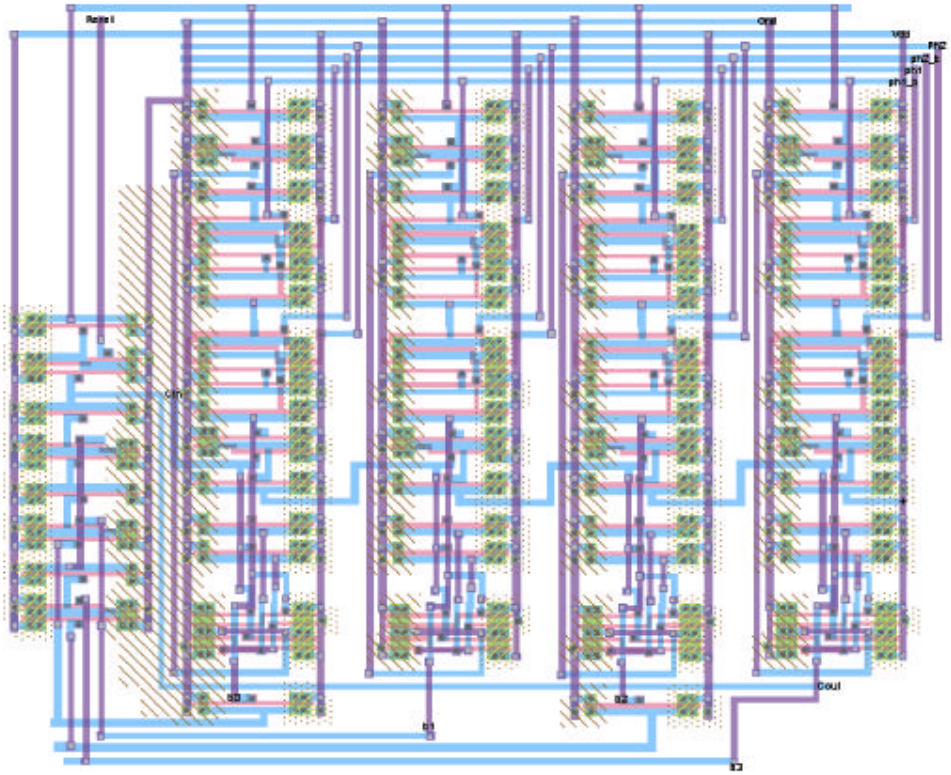


Pulser: Generates one cycle-long impulse from step input

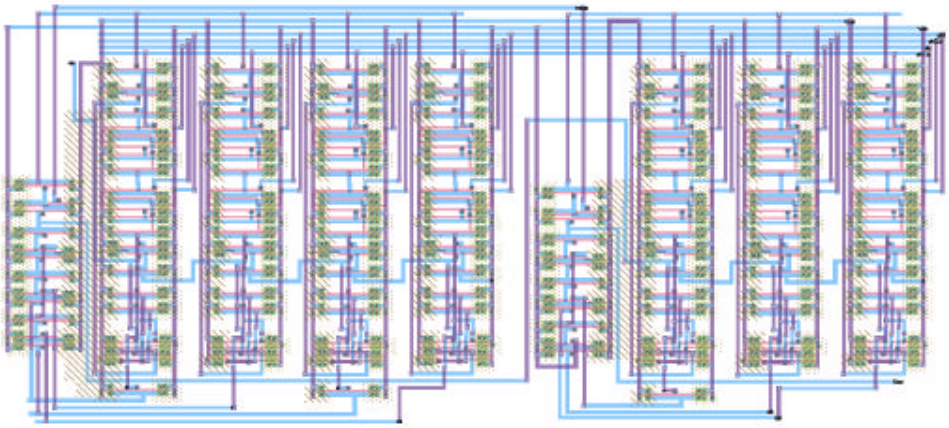
Layout



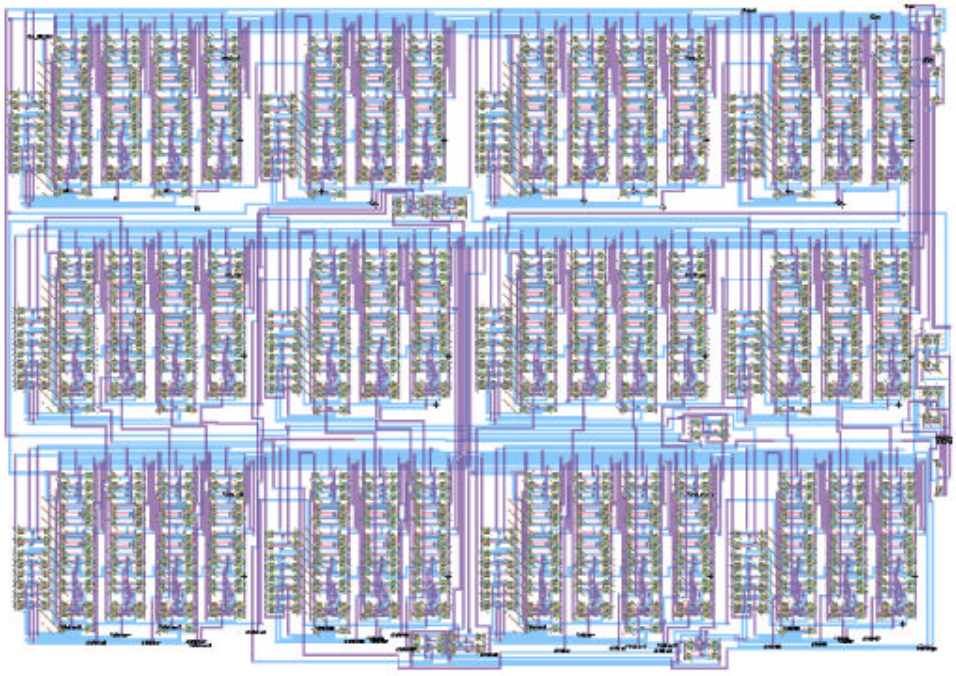
Counter: A one bit synchronous counter



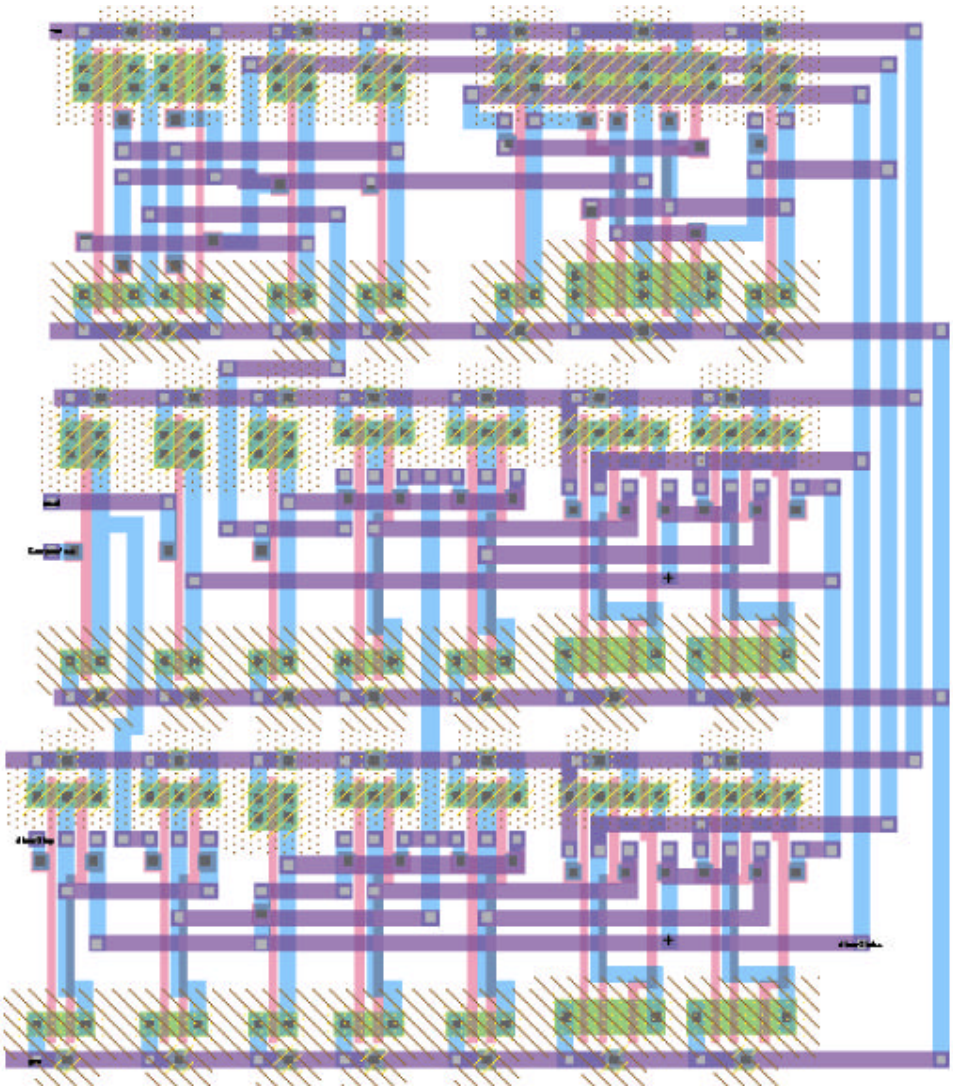
Counter10: A 4 bit counter that resets after accumulating 10



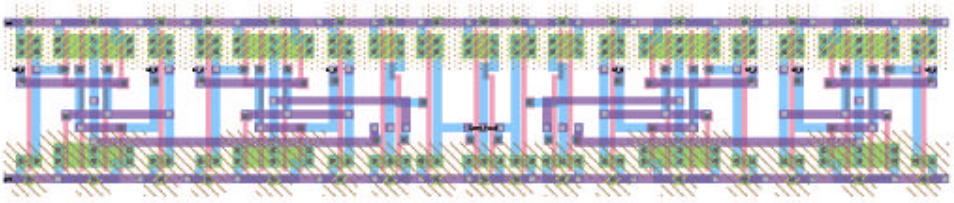
Bitslice: A base 10 counter that resets on 60. Contains one Counter10 and one Counter6



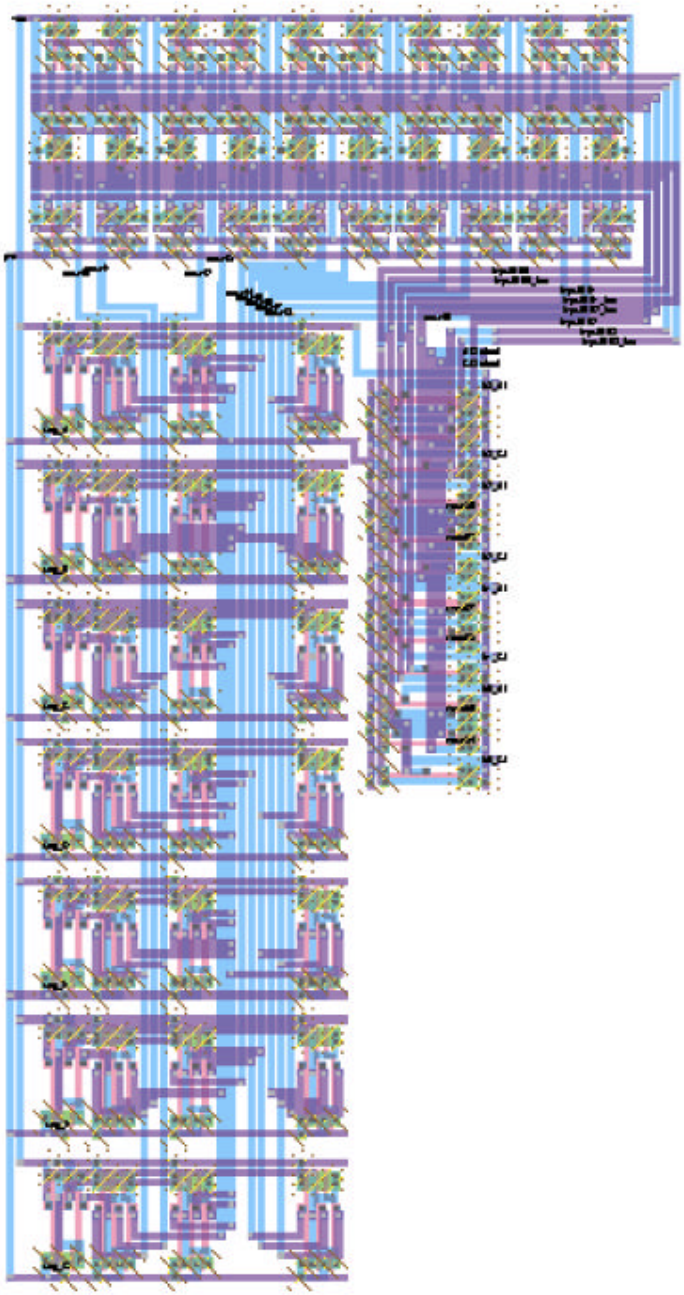
Regfile: The register file. Contains 6 bitslices.



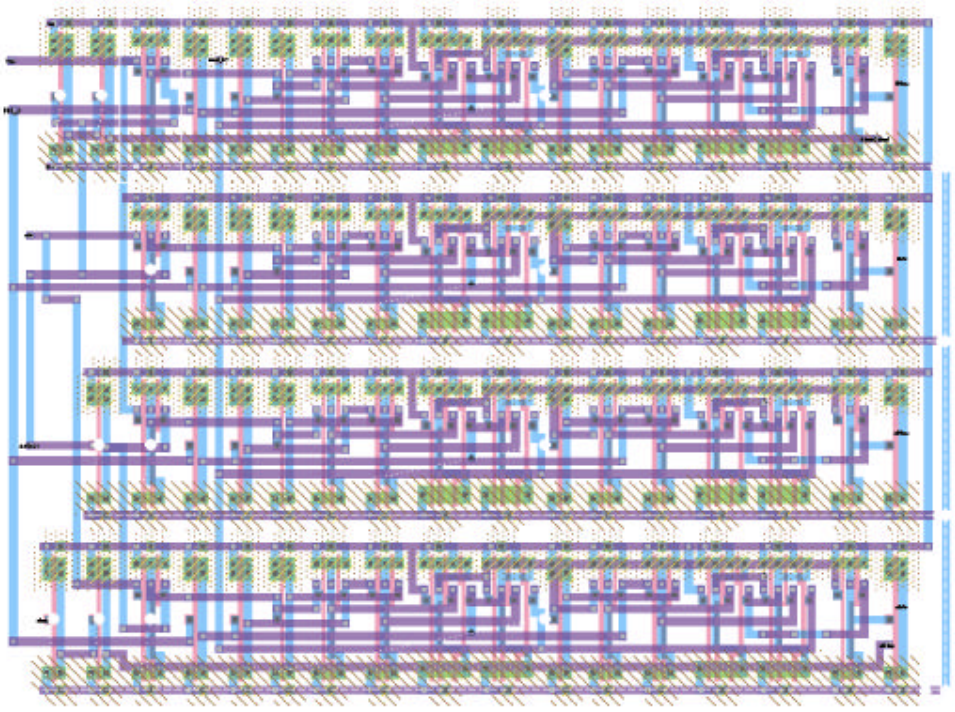
AlarmHold: Holds alarm on until AlarmStop becomes high.



Compare: Compares two four digit numbers.



Decoder: Decodes time for seven-segment display output



Iprocess: Prepares input to be sent to the register file. Contains four Pulsers.