ECSE 323 LAB 5 REPORT

S SQUARED MARS CLOCK

ADITYA SAHA (260453165) & SHAYAN AHMAD (260350431)

Description of System Features

In this lab we integrated all the modules that have been previously built throughout the duration of the course. For making the components functional in the whole integrated system, a new controller module has been implemented. This internally handles all the circuitry required to provide a user interface for the final product.

Features of the integrated system includes:

- Displaying the time on Mars
- Displaying the time on Earth
- Displaying the date on Earth
- Setting time zones on Earth
- Setting time zones on Mars
- Setting time and date on Earth, with added functionality of enabling the Daylight Saving
- Synchronization of Earth and Mars times

User Operation Manual

Altera Controls

- Power button
- Slider switches with unique functionalities, depending on their positional combination-
  - SW0-SW2: mode switches
  - SW3-SW4: display switches (only when in display mode)
  - SW3-SW9: set switches for parameter input (only when in set mode)
- Pushbuttons with functionalities including-
  - KEY3: reset switch
  - KEY2: perform switch
- LED indicator light (LEDR0)

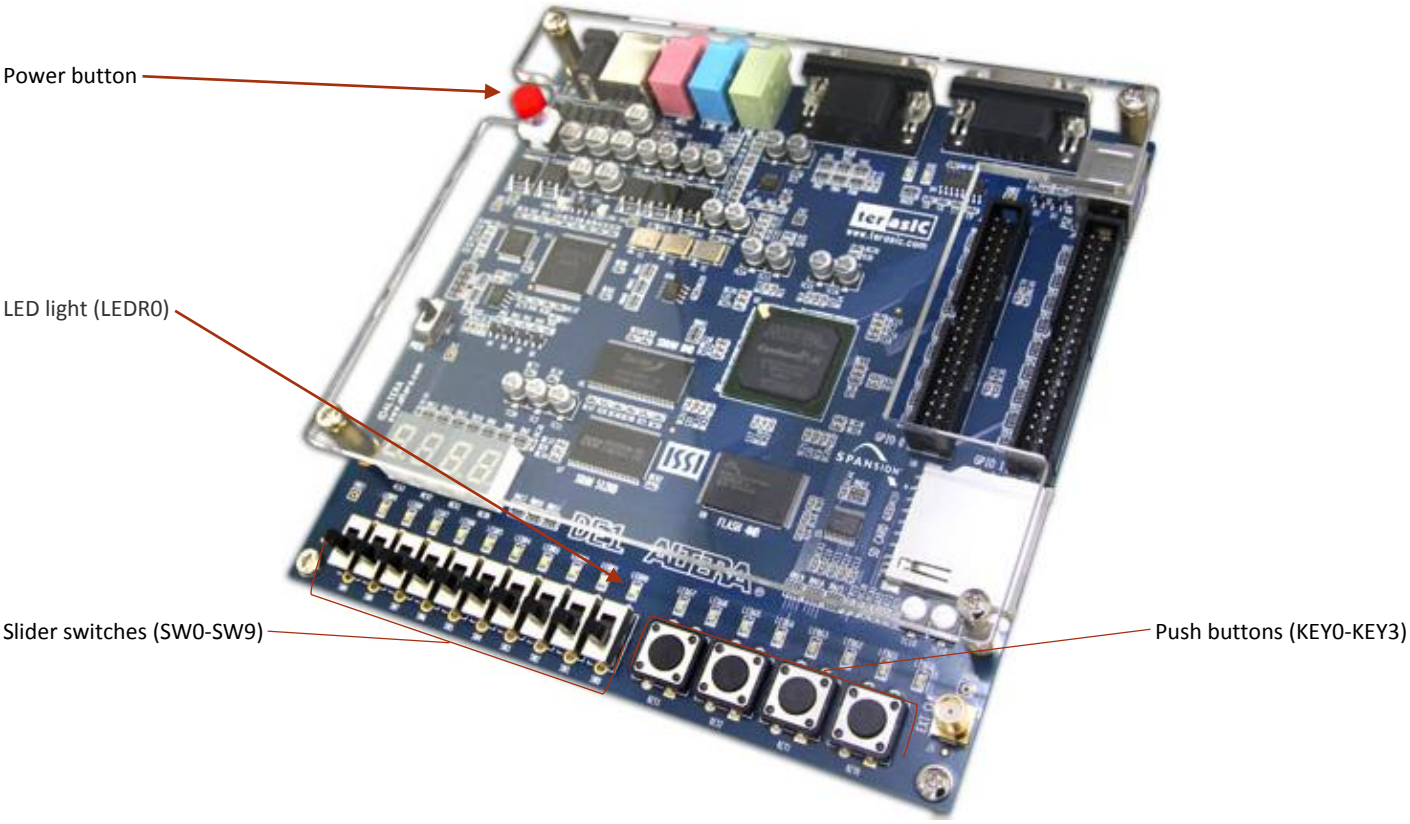

Fig: Schematic for the g20_Basic_Timer circuit (taken from website- http://www.terasic.com.tw/cgi-bin/page/archive.pl?No=83)

Operating Instructions

Quick Guide

- Power button- turns on the system
- Choose mode of operation ***SW2-SW0***
    - Display mode-
        - "***000***" displays ***time*** parameters for ***earth***, also asserts ***LEDR0*** when ***DLS*** is ***ON***
        - "***001***" displays ***date*** parameters for ***earth***
        - "***010***" displays ***time*** parameters for ***mars***
    - Set mode-
        - "***011***" sets ***time zone*** for ***earth***
        - "***100***" sets ***time zone*** for ***mars***
        - "***101***" sets ***earth time*** and ***date***
    - Sync mode-
        - "***110***" ***syncs*** earth and mars ***time***
- Display selection ***SW4-SW3*** (when in display mode)
    - "***00***" displays ***seconds*** or ***days***
    - "***01***" displays ***minutes*** or ***months***
    - "***10***" displays ***hours*** or ***years***
- Parameter set bits ***SW9-SW3*** (when in set mode)

Detailed Walkthrough

The integrated clock system has been ergonomically designed for the most convenient user experience, while making sure that all the available resources (i.e. pushbuttons and slider switches) are economically used to cover all the functionalities pertinent to the provided specification.

Initially after the power button is turned on and the slider switches positioned in rest state, the clock shows seconds count up for earth. As specified in the quick guide, SW2-SW0 initialized as "000" displays the earth clock. Keeping the bits as such, toggling SW4-SW3 displays seconds, minutes and hour parameters for earth as per stated in the guidelines. Similarly, SW2-SW0 initialized as "001" displays the date parameters for earth, and toggling slider switches SW3-SW4 displays the year, month and date parameters. Similar method follows display of time parameters for mars.

The set mode has been implemented as a state sequential method, where assertion of the perform switch (pushbutton KEY2) latches the input bits to the corresponding variable. As an example, to set the date and time parameters for earth, SW2-SW0 is initialized as "101". Then the perform switch is pressed to trigger insert mode. The system then takes bit values (SW9-SW3) for date set followed by perform switch and then month set again followed by perform switch, and goes on to take the respective values and daylight saving assertion finally followed by perform switch being pressed again. An easy set chart is provided in the following to aid the user in the set operation for earth time and date. Note the order of bits is from SW9-SW3, SW3 being the least significant bit in all events. Bit resolutions for each of the input parameters are provided in the circuit description given in the following.

The sync operation follows similar implementation procedure. The buttons SW2-SW0 are initialized to "110" and the perform button is pressed to synchronize the earth and mars clocks. The time zones for earth and mars are also set in similar way.

Note- while setting earth year, a base year of 2000 is considered for ease of simulation. Hence, setting 15 for earth year input bits (i.e. "1111") is equivalent to the year 2015!

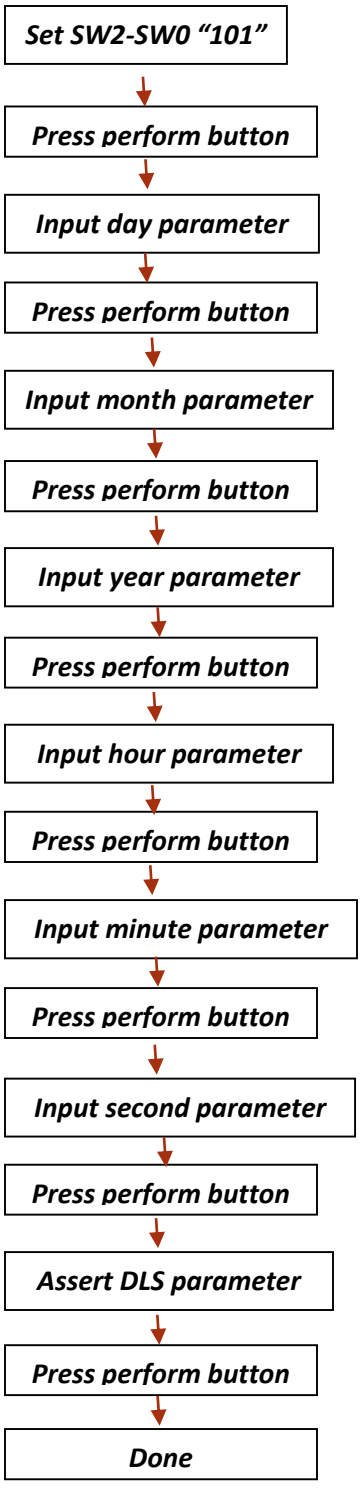

Fig: Flow schematic for set operation of earth year, month, date, hour, minute, second and enable/disable daylight saving

User Interface Implementation

A finite state machine (FSM) has been implemented as an additional component to act as an interface controller. Hence, all the user interface features are managed internally by the circuitry for the FSM. It also acts as an interface to integrate all the previously built modules.

The finite state machine primarily implements the setting functionality of the system. As previously mentioned, the setting follows a sequence of toggling events on the slider switches and intermittent pressing of the ‘perform’ pushbutton.

Starting from the idle state, if the pushbutton is not pressed, the circuit stays in first state- the display state specified as ‘choose1’ (please see vhdl). This state enables us to see the time and date parameters and also check assertion of the daylight savings time (LED signal glows when asserted).

Moving from state ‘choose1’ to the ‘choose2’ requires pressing of the ‘perform’ pushbutton. From this state, the circuit chooses to go to either set-earth-time-zone state with the slider buttons initialized as “011” or set-earth-time-and-date state with the slider buttons initialized as “101” or others (note- state name simplified for understanding, please see vhdl for the actual name of the states). Thus, the state transition follows the bit protocol as previously stated.

Note the use of pushbutton presents a possibility of bouncing effect during operation. This may cause undesired erroneous state transitions, which may necessitate the user to start over every now and again. As a means of debouncing, an intermittent state has been put between every states that requires the user to set bits for a certain parameter. This intermittent state ensures reliable performance of the circuit, ruling out any erroneous possibilities in its operation.

After the final state where the circuit takes assertion bit for daylight saving time, the state transitions goes to the default display state- ‘choose1’. Owing to the controller interface being implemented as a separate standalone module in the integrated system, indicator state enable signals have been declared for each state (‘mars_hms_load_en’ for e.g.), which gets asserted when the circuit operates in that respective state. In the master vhdl code for the complete system, these enable signals are used for implementing the conditional operations when the circuit resides in a certain state.

Please see vhdl for more clarity on the states and variables referred to in the above explanation. A snippet of the state diagram is included in the ending pages of the documentation.

System Design

A block diagram of the complete integrated system is included in the following to aid the reader better understanding of the system implementation. The inputs and outputs to the system are defined in the following-

- 1-bit clock input, specified as ‘clk’
- 1-bit reset input, specified as ‘rst’
- 1-bit pushbutton input, specified as ‘perform’
- 10-bit slider switch input, specified as ‘switches’
- 1-bit indicator output, specified as ‘dst_out’
- 4 7-bit LED display outputs, specified as ‘hex(n)[6..0]’

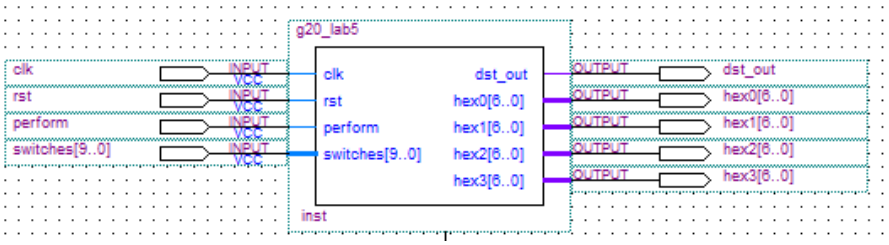

Fig: Higher abstraction for the integrated circuit

The clock input is mapped to the 50MHz clock on the Altera board. It has been noted however, on performing multiple simulations, that the circuit never attains 50MHz frequency. In fact the circuit has been seen to attain a maximum of 45.55MHz. While this does not cause occurrence of error in regular operation, this is however a discrepancy between the circuit’s performance specification and actual performance attained.

Timing Analyzer Summary

	Type	Slack	Required Time	Actual Time
1	Worst-case tsu	N/A	None	9.116 ns
2	Worst-case tco	N/A	None	22.198 ns
3	Worst-case tpd	N/A	None	22.341 ns
4	Worst-case th	N/A	None	0.650 ns
5	Clock Setup: 'clk'	N/A	None	45.41 MHz (period = 22.024 ns)
6	Total number of failed paths			

Fig: Timing Analyzer Summary for the integrated circuit showing clock frequency max out at 45.41MHz

As previously mentioned, the controller module implements the user interface of the system. The controller module is imported into the master vhdl code (g20_lab5.vhd), which only implements the display functionality of the system. Only single process block is implemented, which takes into the sensitivity list the previously mentioned state enable signals among other signals. This ensures that the system keeps track of the bit manipulation done on the input signals and operates conditionally case-by-case.

Testing

Finite State Machine Testing

Case- Test appropriate state transitions following provided inputs

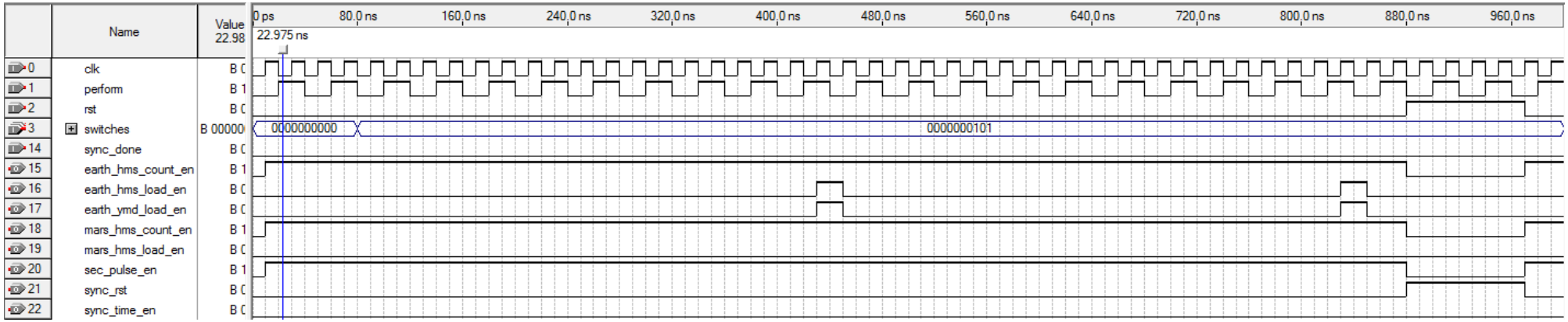

Fig: Functional simulation of the FSM testing appropriate state transitions

Full System Testing

Case- Test display

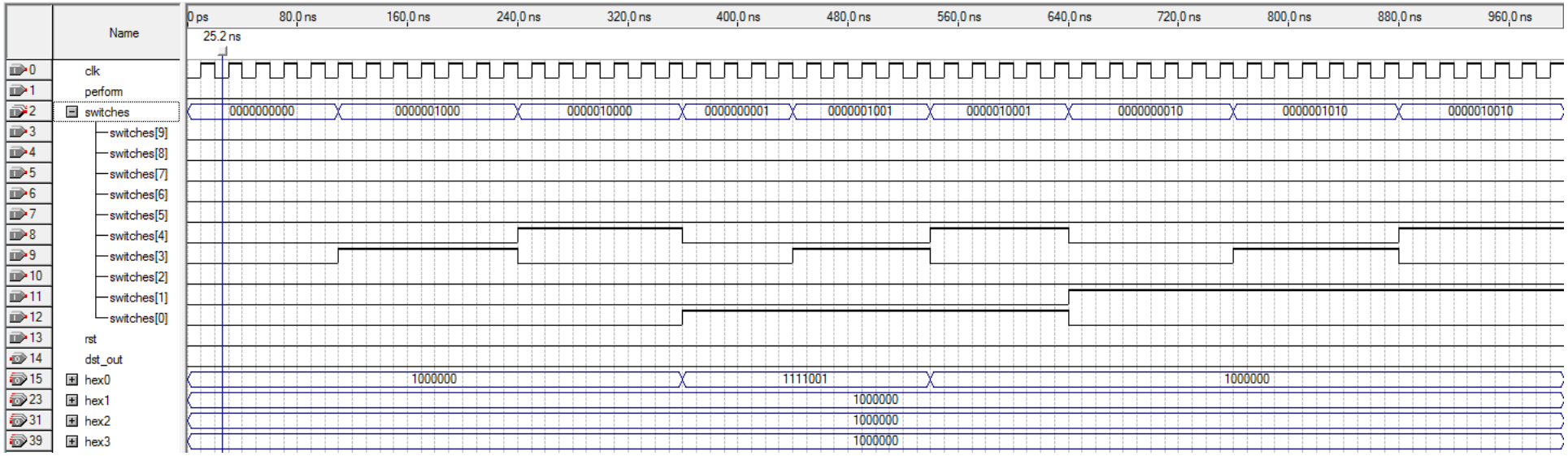

Fig: Functional simulation of the integrated system checking LED display for respective switch inputs

Case- Test input latching

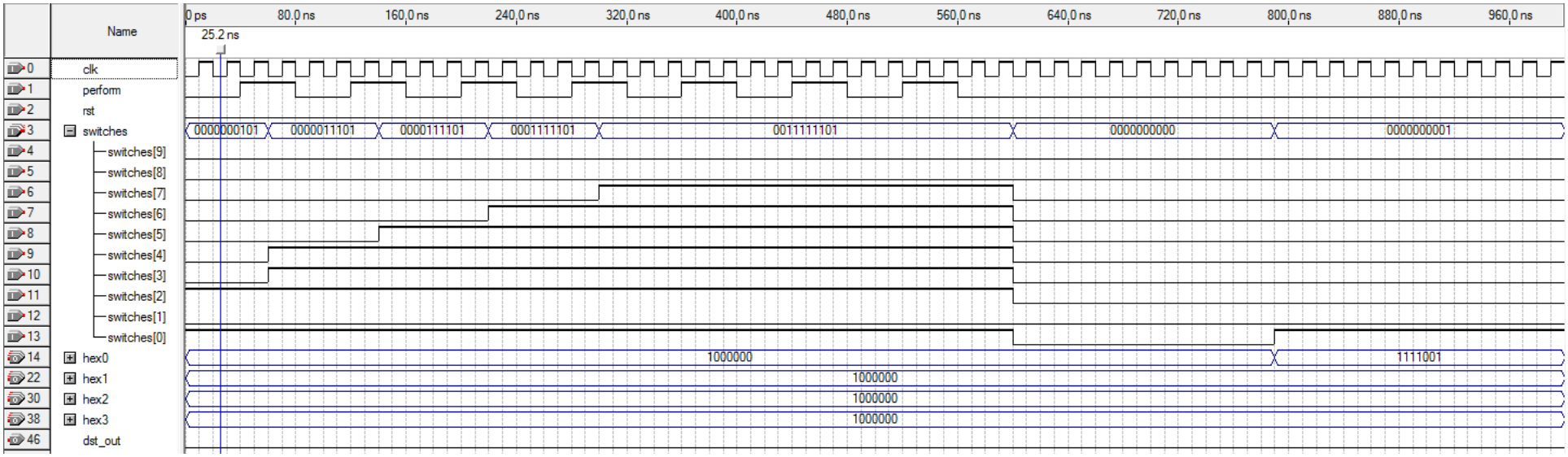

Fig: Functional simulation of the integrated system checking LED display for proper latching of inputs followed by LED display

Case- Altera onboard test

The following pin assignments have been used to test the circuit on board.

	Node Name	Direction	Location	I/O Bank	VREF Group	I/O Standard	Reserved
1	clk	Input	PIN_L1	2	B2_N1	3.3-V LVTTL (default)	
2	dst_out	Output	PIN_R20	6	B6_N0	3.3-V LVTTL (default)	
3	hex0[6]	Output	PIN_E2	2	B2_N1	3.3-V LVTTL (default)	
4	hex0[5]	Output	PIN_F1	2	B2_N1	3.3-V LVTTL (default)	
5	hex0[4]	Output	PIN_F2	2	B2_N1	3.3-V LVTTL (default)	
6	hex0[3]	Output	PIN_H1	2	B2_N1	3.3-V LVTTL (default)	
7	hex0[2]	Output	PIN_H2	2	B2_N1	3.3-V LVTTL (default)	
8	hex0[1]	Output	PIN_J1	2	B2_N1	3.3-V LVTTL (default)	
9	hex0[0]	Output	PIN_J2	2	B2_N1	3.3-V LVTTL (default)	
10	hex1[6]	Output	PIN_D1	2	B2_N0	3.3-V LVTTL (default)	
11	hex1[5]	Output	PIN_D2	2	B2_N0	3.3-V LVTTL (default)	
12	hex1[4]	Output	PIN_G3	2	B2_N0	3.3-V LVTTL (default)	
13	hex1[3]	Output	PIN_H4	2	B2_N0	3.3-V LVTTL (default)	
14	hex1[2]	Output	PIN_H5	2	B2_N0	3.3-V LVTTL (default)	
15	hex1[1]	Output	PIN_H6	2	B2_N0	3.3-V LVTTL (default)	

Fig: Pin assignment table for Altera board testing (part 1 of 2)

16		hex1[0]	Output	PIN_E1	2	B2_N1	3.3-V LVTTL (default)	
17		hex2[6]	Output	PIN_D3	2	B2_N0	3.3-V LVTTL (default)	
18		hex2[5]	Output	PIN_E4	2	B2_N0	3.3-V LVTTL (default)	
19		hex2[4]	Output	PIN_E3	2	B2_N0	3.3-V LVTTL (default)	
20		hex2[3]	Output	PIN_C1	2	B2_N0	3.3-V LVTTL (default)	
21		hex2[2]	Output	PIN_C2	2	B2_N0	3.3-V LVTTL (default)	
22		hex2[1]	Output	PIN_G6	2	B2_N0	3.3-V LVTTL (default)	
23		hex2[0]	Output	PIN_G5	2	B2_N0	3.3-V LVTTL (default)	
24		hex3[6]	Output	PIN_D4	2	B2_N0	3.3-V LVTTL (default)	
25		hex3[5]	Output	PIN_F3	2	B2_N0	3.3-V LVTTL (default)	
26		hex3[4]	Output	PIN_L8	2	B2_N1	3.3-V LVTTL (default)	
27		hex3[3]	Output	PIN_J4	2	B2_N1	3.3-V LVTTL (default)	
28		hex3[2]	Output	PIN_D6	2	B2_N0	3.3-V LVTTL (default)	
29		hex3[1]	Output	PIN_D5	2	B2_N0	3.3-V LVTTL (default)	
30		hex3[0]	Output	PIN_F4	2	B2_N0	3.3-V LVTTL (default)	
31		perform	Input	PIN_T22	6	B6_N0	3.3-V LVTTL (default)	
32		rst	Input	PIN_T21	6	B6_N0	3.3-V LVTTL (default)	
33		switches[9]	Input	PIN_L2	2	B2_N1	3.3-V LVTTL (default)	
34		switches[8]	Input	PIN_M1	1	B1_N0	3.3-V LVTTL (default)	
35		switches[7]	Input	PIN_M2	1	B1_N0	3.3-V LVTTL (default)	
36		switches[6]	Input	PIN_U11	8	B8_N0	3.3-V LVTTL (default)	
37		switches[5]	Input	PIN_U12	8	B8_N0	3.3-V LVTTL (default)	
38		switches[4]	Input	PIN_W12	7	B7_N1	3.3-V LVTTL (default)	
39		switches[3]	Input	PIN_V12	7	B7_N1	3.3-V LVTTL (default)	
40		switches[2]	Input	PIN_M22	6	B6_N0	3.3-V LVTTL (default)	
41		switches[1]	Input	PIN_L21	5	B5_N1	3.3-V LVTTL (default)	
42		switches[0]	Input	PIN_L22	5	B5_N1	3.3-V LVTTL (default)	

Fig: Pin assignment table (contd.) for Altera board testing (part 2 of 2)

Description of Modules Used

This lab integrates all the modules that have been previously built throughout the duration of the course. While all of them have been implicitly instantiated at one point or other, some of the major modules that have been implemented in the master data path module and their functional description is provided in the following.

g20_YMD_Counter

A circuit that counts up Day, Month and Year for Earth and resets according to specification.

The circuit is required to count up for days, months and years on Earth. It is expected to do so as per regular calendar counting- which necessitated the number of days in a month and the leap year considerations to be made during the circuit implementation. The circuit implements process block with the asynchronous clock and reset signals passed to the sensitivity list. Before implementing the process block, explicit cases for special situations have been outlined. This includes consideration of months with 30 and 31 days, and also leap year. Cases have also been defined for day, month and year reset. The circuit is required to count up to the year 4000.

The main logic of the circuit is implemented in the process block. The block contains asynchronous clock and reset signals passed to its sensitivity list. The circuit is rising edge triggered, implying that changes will only take place during the rising edge of the clock transitions. Following the explicit case definitions made before the declaration of the process block and the commenting made all throughout, the vhdl logic implementation within the block is self-explanatory.

The g20_YMD_Counter circuit has inputs and outputs defined as follows:

- 1-bit asynchronous clock input, specified as ‘clock’
- 1-bit asynchronous reset input, specified as ‘reset’
- Two 1-bit enable inputs, specified as ‘day_count_en’ and ‘load_enable’
- Three load inputs named ‘Y_Set’= 12-bit, ‘M_Set’= 4-bit and ‘D_Set’= 5-bit
- Three outputs named ‘Years’= 12-bit, ‘Months’= 4-bit and ‘Days’= 5-bit

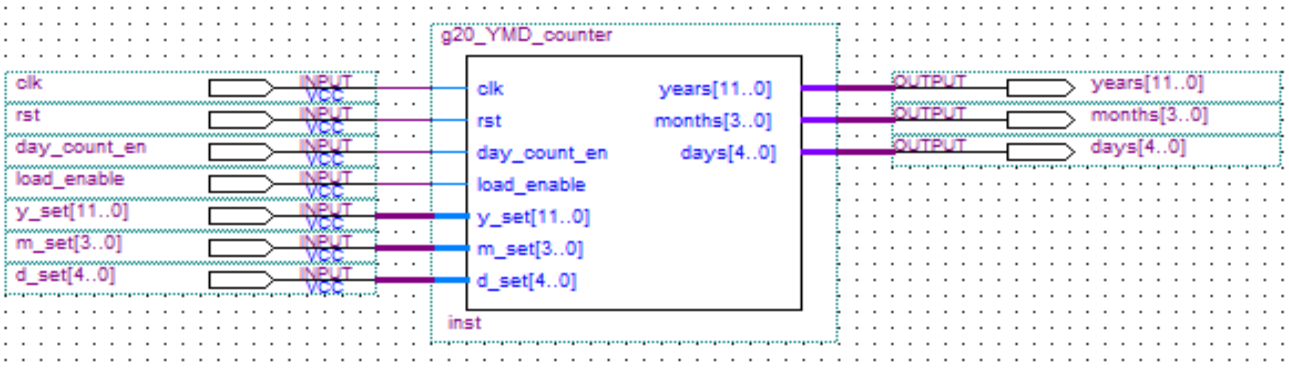

Fig: Higher abstraction for g20_YMD_counter circuit

g20_UTC_to_MTC

A circuit that takes in Earth time and date and generates the corresponding Mars time of day.

The circuit is required to calculate the time of day in Mars, given time and date parameters on Earth. The motive behind this circuit implementation is to synchronize both the Earth and Mars clock, i.e. set the Mars clock following the Earth parameters. The implementation of this circuit largely relies on the standardisation discussions and conversion methods stated in the lab slides. The computation of the Mars time involves other modular computations in the process. Notably using the formula given in the lab slides, ‘NDays’ is computed in a process block that takes asynchronous clock and reset signals in the sensitivity list. This variable represents the number of times the YMD counter gets incremented. Another variable ‘day_frac’ which is the day-fraction representation of seconds is computed using one of the previously built lab modules (g20_Seconds_to_Days). Following computation of each of these variables the corresponding Julian Date is

evaluated. The circuit finally implements LPM multiplication and add-sub module to extract the hours, minutes and seconds components of MTC, followed by a process block implementation to latch the extracted outputs. The computation primarily employs concepts of fixed point arithmetic. A 1-bit output ‘done’ is asserted when the operation completes.

The g20_UTC_to_MTC circuit has inputs and outputs defined as follows:

- 1-bit asynchronous clock input, specified as ‘clock’
- 1-bit asynchronous reset input, specified as ‘reset’
- Three load inputs named ‘earth_year’= 12-bit, ‘earth_month’= 4-bit and ‘earth_day’= 5-bit
- Three load inputs named ‘earth_hour’= 5-bit, ‘earth_min’= 6-bit and ‘earth_sec’= 6-bit
- Three load outputs named ‘mars_hour’= 5-bit, ‘mars _min’= 6-bit and ‘mars _sec’= 6-bit
- 1-bit output named ‘done’

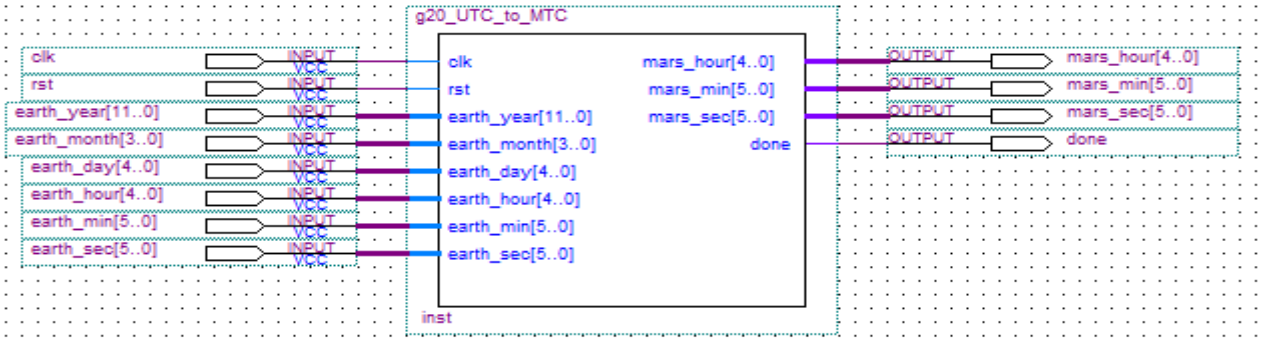

Fig: Higher abstraction for the g20_UTC_to_MTC circuit

g20_Basic_Timer

As circuit that acts as a frequency divider and provides an output pulse at each second.

The circuit is implemented following the schematic provided in the lab slides. An LPM down counter module is used to count down from a constant value and set the counter output signal high once the count reaches 0. The circuit is sequential, hence the output has been fed back to the input in order to reset the counter (start over counting down to 0). Hence, counter reset is asserted by implementing OR operation of the ‘reset’ input signal with the counter output/feedback signal. An LPM constant module has been used to feed in a constant value (of specific bits) to the circuit (the constant value to count down from). The calculation for the constant value is given below.

The g20_Basic_Timer has three inputs and two outputs:

- 1-bit clock input, specified as ‘clock’
- 1-bit counter reset input, specified as ‘reset’
- 1-bit counter enable input, specified as ‘enable’
- Two 1-bit outputs named EPULSE and MPULSE

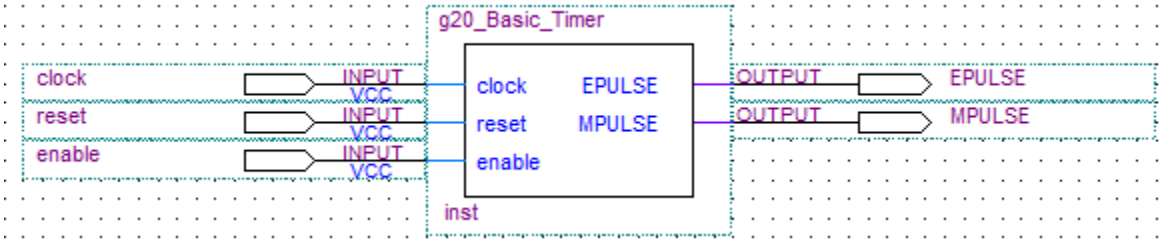

Fig: Higher abstraction for the g20_Basic_Timer circuit

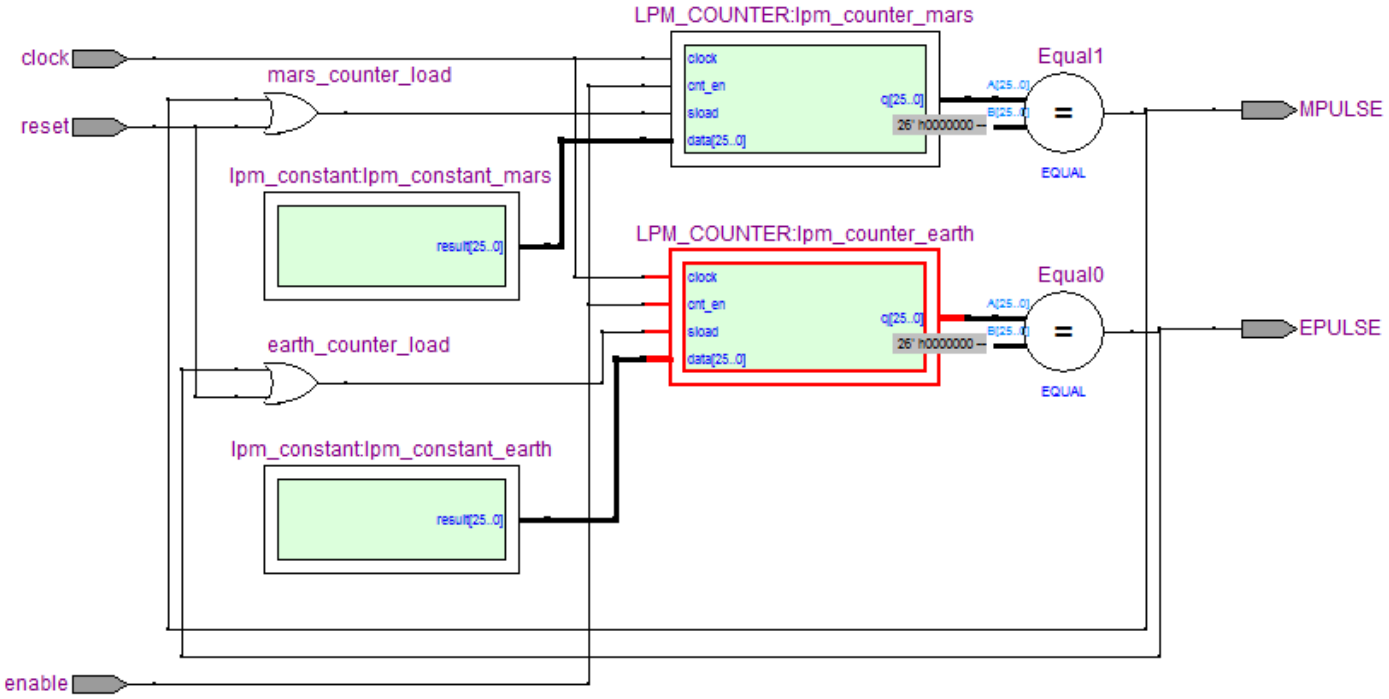

Fig: Schematic for the g20_Basic_Timer circuit

g20_HMS_Counter

A circuit that counts up hour, minute and second for Earth and Mars.

The circuit implemented is a counter circuit that counts up hours, minutes and seconds parameters for both Earth and Mars, depending on the unit second pulse being fed to the circuit (i.e. EPULSE or MPULSE). LPM up counter module has been used in the circuit implementation. The circuit also has the functionality of setting the hour, minute and second parameter to a desired value.

The g20_HMS_Counter circuit has 8 inputs and 3 outputs:

- 1-bit clock input, specified as ‘clock’
- 1-bit reset input, specified as ‘reset’
- 1-bit second pulse input, specified as ‘sec_clock’
- 1-bit enable inputs, specified as ‘count_enable’ and ‘load_enable’
- 1 5-bit and 2 6-bit load inputs respectively specified as ‘H_Set’, ‘M_Set’ and ‘S_Set’
- 1 5-bit and 2 6-bit outputs respectively specified as ‘Hours’, ‘Minutes’ and ‘Seconds’
- 1-bit indicator output, specified as ‘end_of_day’

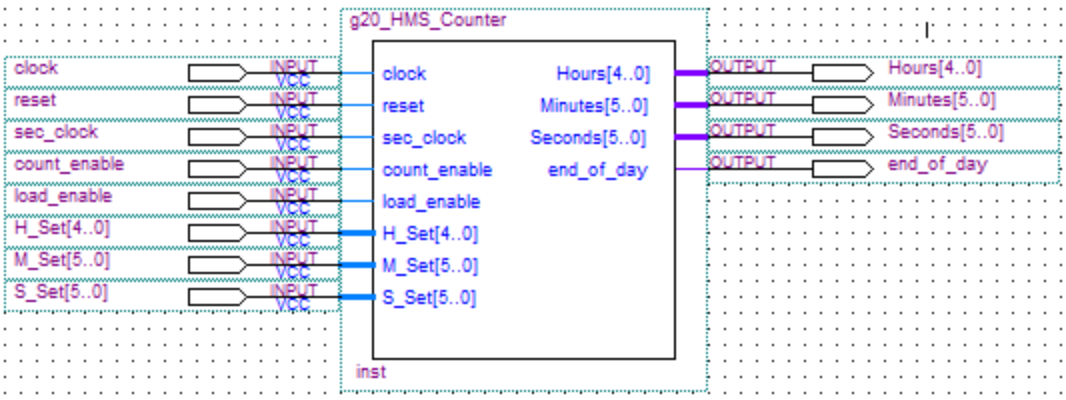

Fig: Higher abstraction for the g20_HMS_Counter circuit

g20_7_segment_decoder

A circuit that implements 7-segment LED decoder.

The circuit implemented is a decoder circuit that takes in a 4 bit code and one ripple bit and converts to 7 bit representation to be used for display purpose on the LED display of the Altera board.

The g20_7_segment_decoder has 2 inputs and 2 outputs:

- 4-bit input, specified as ‘code’
- 1-bit ripple input, specified as ‘RippleBlank_In’
- 1-bit ripple output, specified as ‘RippleBlank_Out’
- 7-bit LED display output, specified as ‘segments’

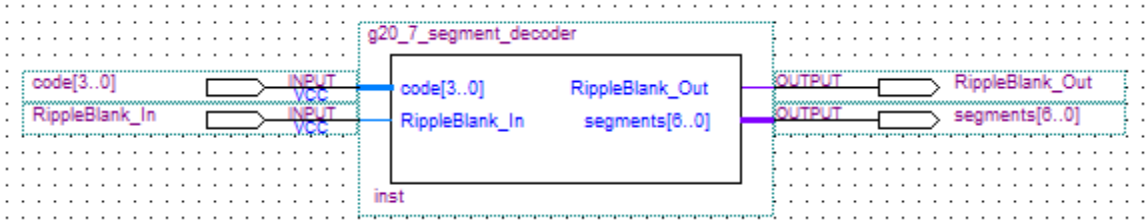

Fig: Higher abstraction for the g20_7_segment_decoder circuit

FPGA Summary and Conclusion

The setting of time zones for earth and mars seems to be erroneous for some reason. Due to shortage of time budget, they are left as is. As previously mentioned already, the clock frequency requirement has not met the specification, which is a discrepancy between the proposed and actual implementation of the system.

The feature that required most of the time and effort was designing the user interface of the system. Given the resources we are provided with are somewhat limited (i.e. the buttons and slider switches) in comparison to the number of parameters we are required to set, we had to think out of the box for the whole vhdl implementation to deliver the most ergonomic system.

As already stated, if there was little more time budget more effort would have been expended on full system functional testing.

Flow Status	Successful - Mon Apr 14 03:20:19 2014
Quartus II 64-Bit Version	9.1 Build 350 03/24/2010 SP 2 SJ Full Version
Revision Name	g20_lab5
Top-level Entity Name	g20_lab5
Family	Cyclone II
Device	EP2C20F484C7
Timing Models	Final
Met timing requirements	Yes
Total logic elements	970 / 18,752 (5 %)
Total combinational functions	952 / 18,752 (5 %)
Dedicated logic registers	293 / 18,752 (2 %)
Total registers	293
Total pins	42 / 315 (13 %)
Total virtual pins	0
Total memory bits	0 / 239,616 (0 %)
Embedded Multiplier 9-bit elements	18 / 52 (35 %)
Total PLLs	0 / 4 (0 %)

Fig: FPGA resource utilization summary for the circuit

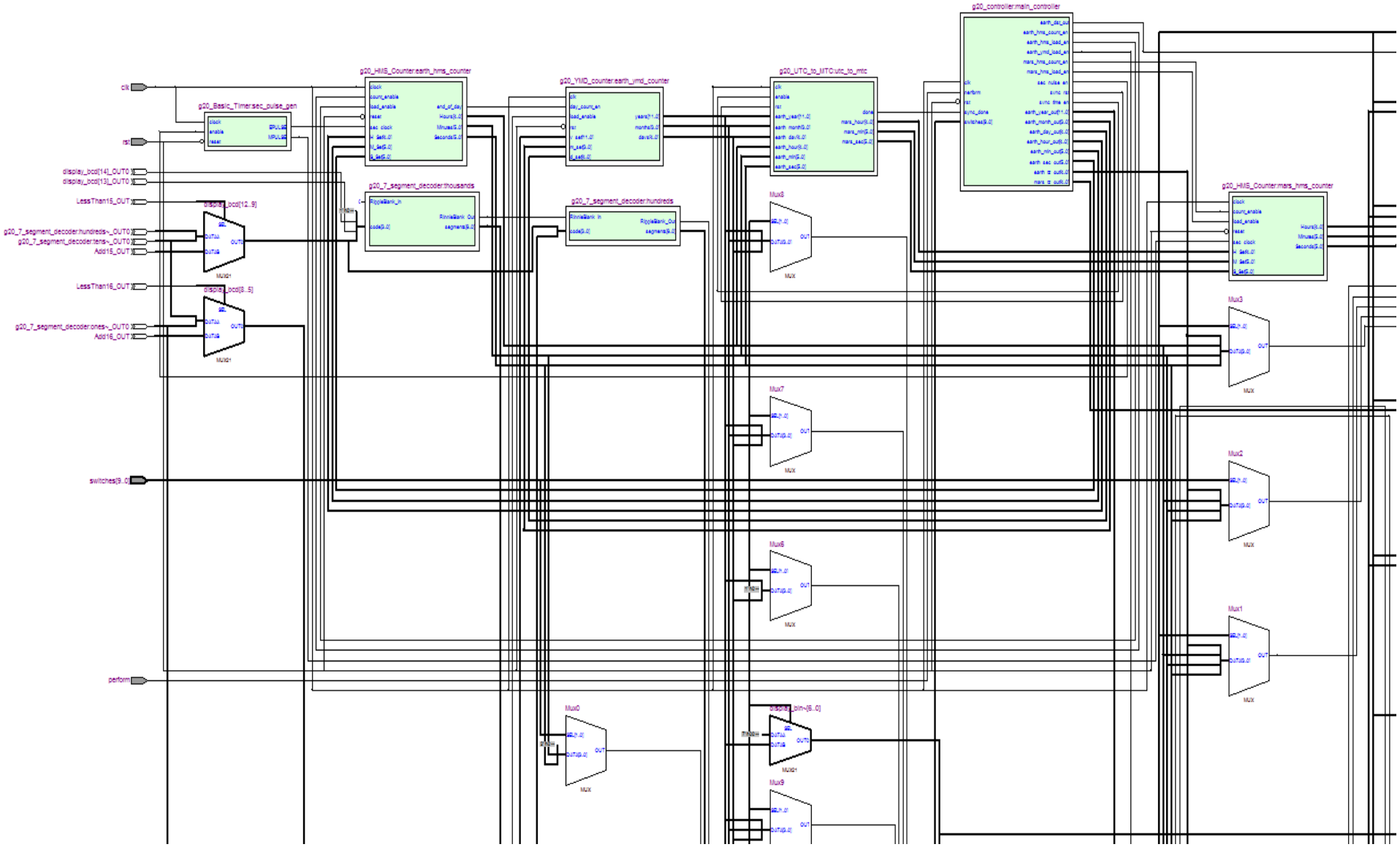

Fig: Schematic of the integrated circuit (part 1 of 4)

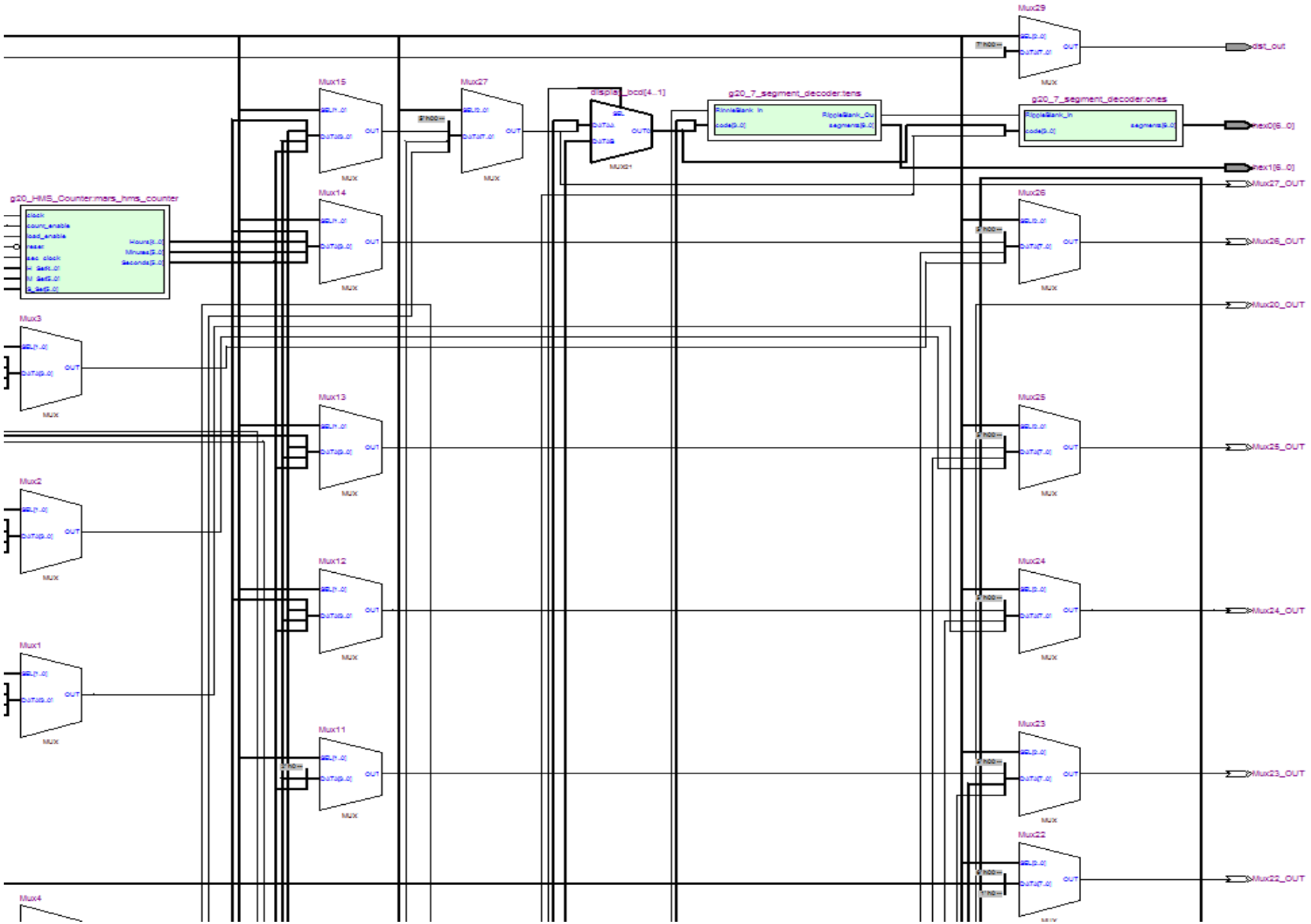

Fig: Schematic of the integrated circuit (part 2 of 4)

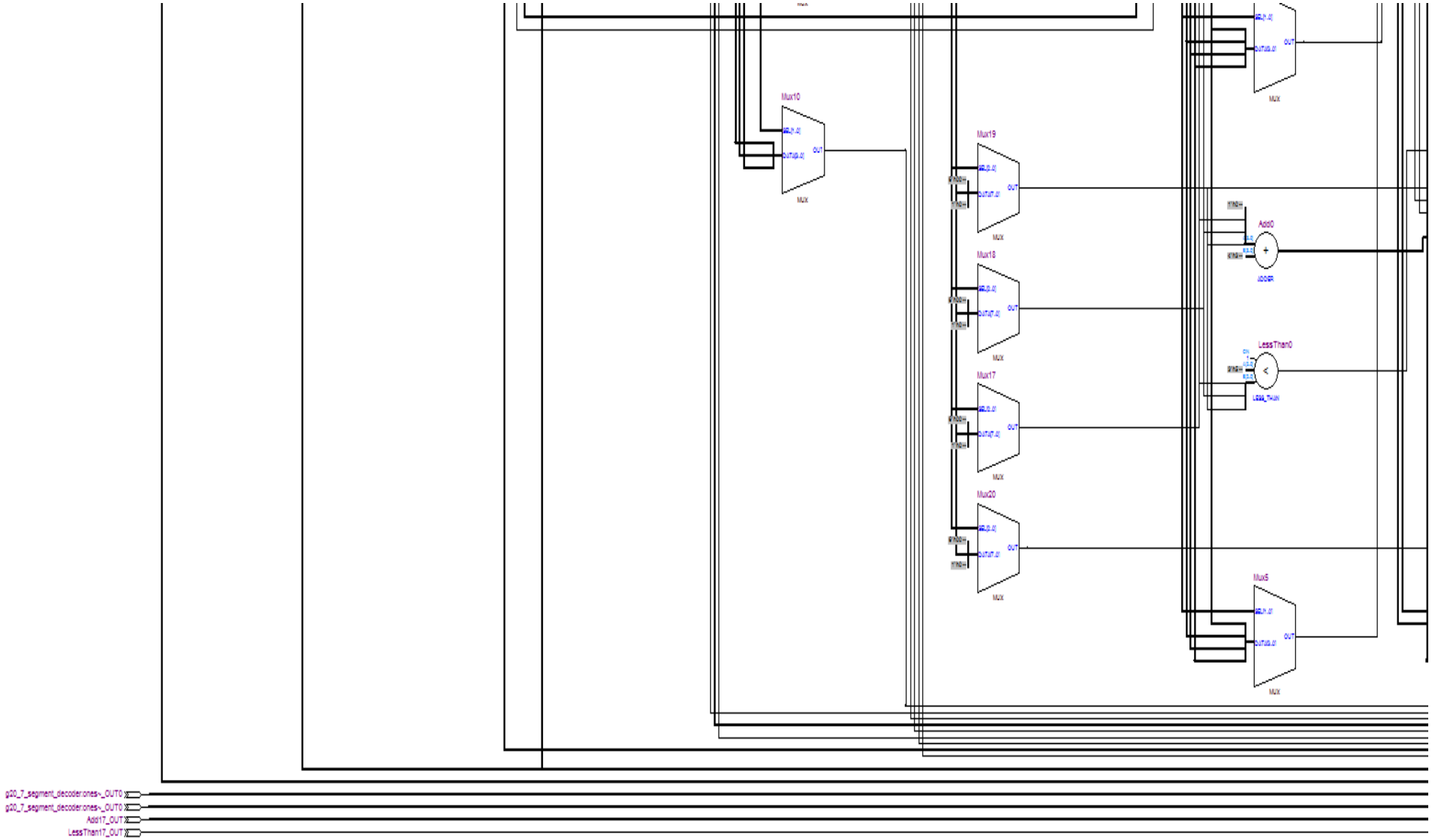

Fig: Schematic of the integrated circuit (part 3 of 4)

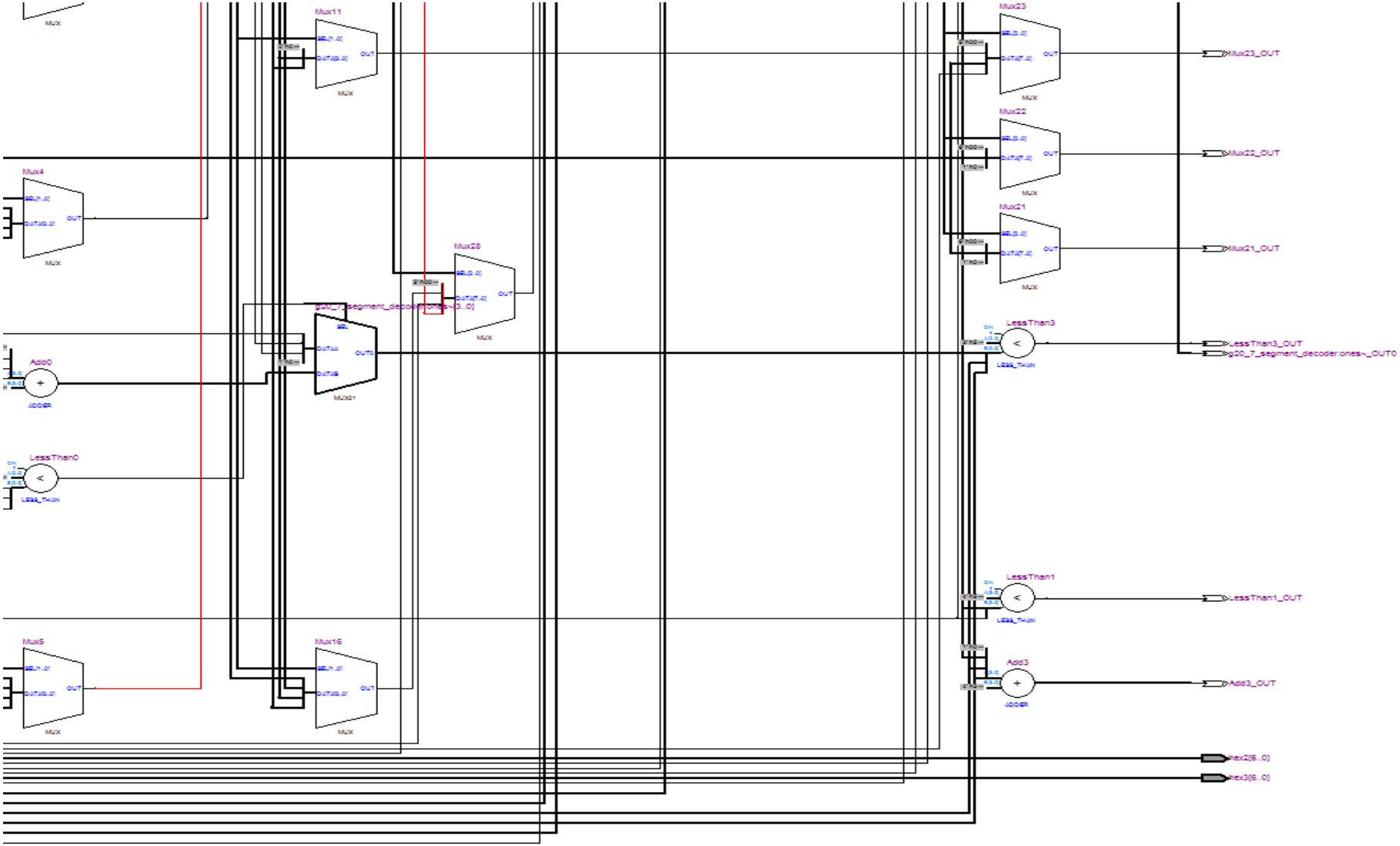

Fig: Schematic of the integrated circuit (part 4 of 4)

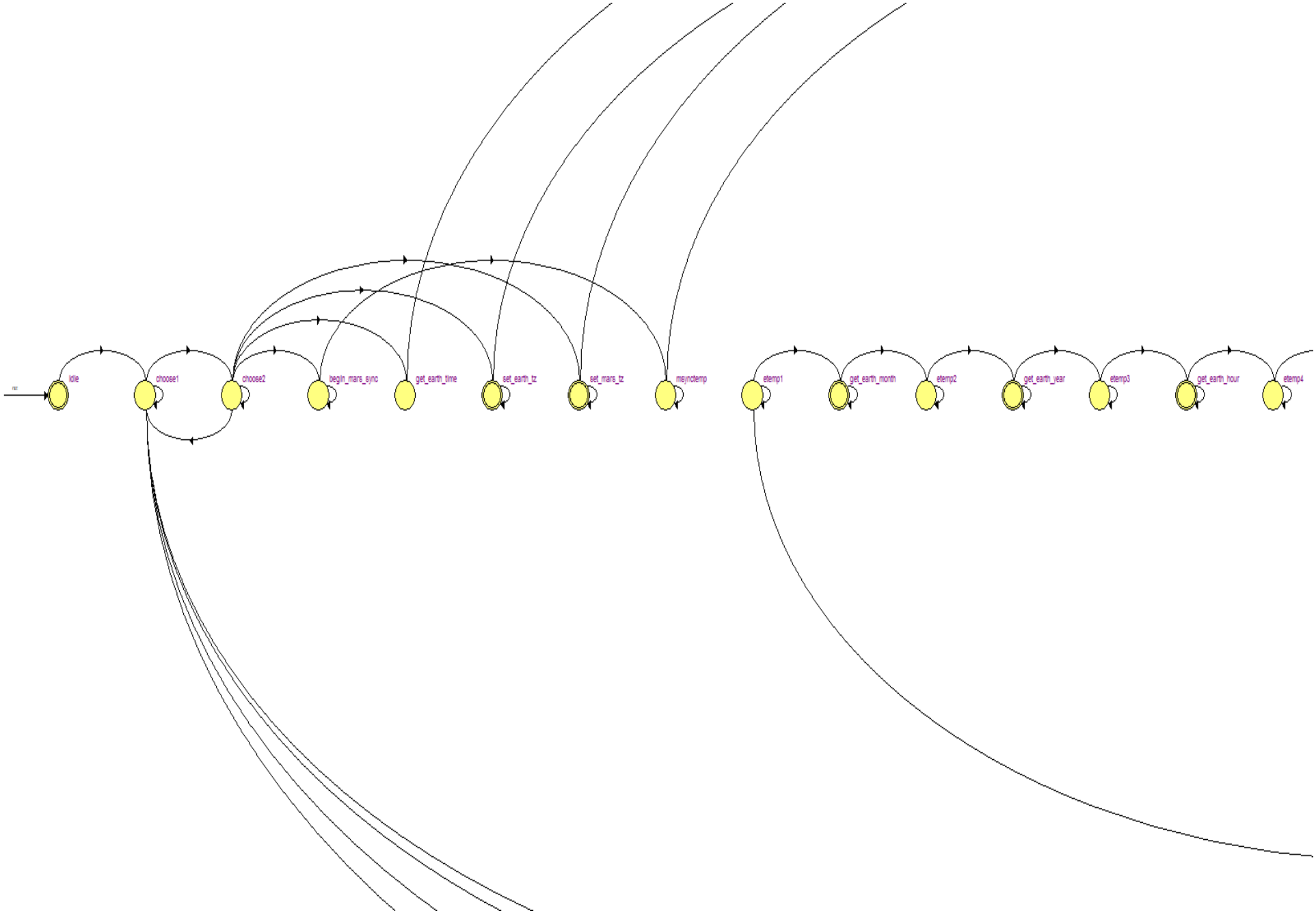

Fig: Schematic of the state transition (part 1 of 2)

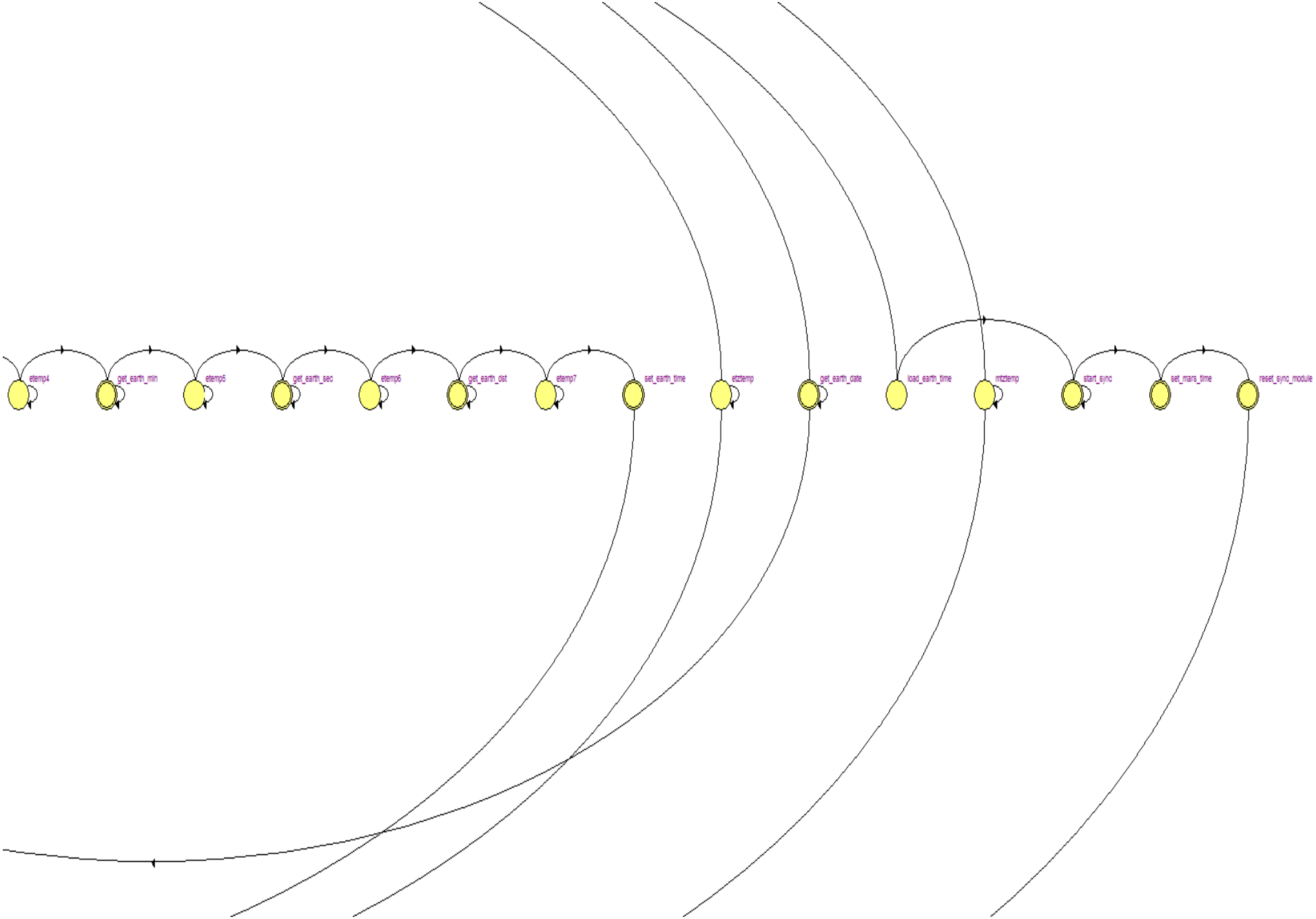

Fig: Schematic of the state transition (part 2 of 2)