Cadence Design of clock/calendar using 240*8 bit RAM using Verilog HDL

K.R.N.Karthik [#1], M.Nagesh Babu [#2], Fazalnoorbasha [#3]

[#1]Student of VLSI Systems Research Group, Department of Electronics and Communication Engineering, K.L University, Guntur, AP – INDIA

[2]ASSOC. Professor VLSI Systems Research Group, Department of Electronics and Communication Engineering, K.L University, Guntur, AP – INDIA

[3]ASSOC. Professor VLSI Systems Research Group, Department of Electronics and Communication Engineering, K.L University, Guntur, AP – INDIA

[1]karthik.kadava@kluniversity.in [2]m.nageshbabu@kluniversity.in [3]fazalnoorbasha@kluniversity.in

***Abstract*—** In the contrast of the modern day technology evolution the number of electronic components increasing on a system. New electronic control units (ECUs) are not only dedicated to entertainment, but also for increasing safety and comfort. More and more mechanical connections are replaced by electronic ones to save energy and increase comfort and security. All these electronic devices need a way of exchanging information on a fast, reliable and robust way. As there was a tremendous change in the technology day by day mainly in the field of chip designing and the automation technology as due to this the clock speeds are also rapidly increasing along with this power measures are also increasing so to manage this situation we are moving towards the clock/calendar. The clock/calendar circuit based on 2048-bit static RAM organized as 256 words by 8 bits .Address and data are transferred serially via the two-line bidirectional I2C-bus The built in word address register is incremented automatically after each written of read data byte .Addressing pin A0 is used for programming the hard ware address .allowing the connection of two device to bus without additional hardware This total module can be used as a real time clock of adjustable frequencies and can also replace the purpose of the counters on the digital based applications This is designed in verilog using Xilinx and cadence 90nm in LINUX environment

Keywords-Automation technology, Digital Circuit, RAM.

I. Introduction

The **CLOCK/CALENDAR** based on a 2048 bit static RAM organized as 256 words by 8 bits. Addresses and data are transferred serially via the two-line bidirectional I2C-bus [1]. The built-in word address register is incremented automatically after each written or read data byte. Address pin A0 is used for programming the hardware address, allowing the connection of two devices to the bus without additional hardware. first 8 bytes of the RAM are used for the clock, calendar, and counter functions. The next 8 bytes can be programmed as alarm registers or used as free RAM space. The remaining 240 bytes are free RAM locations.

I am using the I2C protocol for the serial transmission of the data and also for the communication of this clock or calendar with various digital circuitry on a chip serial clock is generated by the master and SDA, bidirectional line is for address and data transmission The I2C bus is built around a two-wire serial bus, SDA (serial data) and SCL (serial clock). Each device is recognized by a unique address, and can operate either as a transmitter or a as a receiver. The I2C master is the device that initiates a transfer and generates the clock for the same. Any device addressed by the master is the slave. If more than one master attempts to transmit in unison, there will be a conflict. The I2C specification solves this conflict by its arbitration process. When the master (your controller) wishes to talk to a slave. It begins by issuing a start sequence on the I2C bus. A start sequence is one of two special sequences defined for the I2C bus, the other being the stop sequence. The start sequence and stop sequence are special in that these are the only places where the SDA (data line) is allowed to change while the SCL (clock line) is high. When data is being transferred, SDA must remain stable and not change whilst SCL is high. The start and stop sequences mark the beginning and end of a transaction with the slave device.

II. PROPOSED BLOCK DIAGRAM

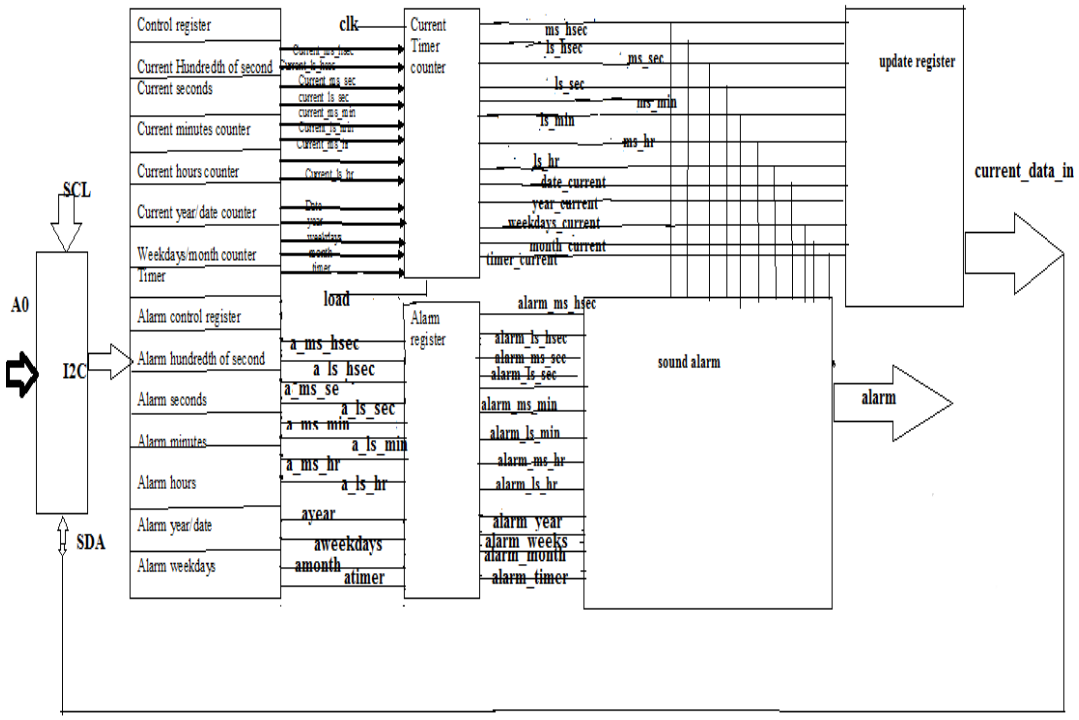

Fig 1: Block diagram of clock/calendar

The block diagram consists of six modules such as RAM, I2C [1], counter, Alarm register, Sound alarm, Update register and also consists of SDA and SCL line.

A. *Counter*

The counter [3] module is to count the present time and the count depends on the seconds pulse, and all other parameters such as the minutes, hours, days and calendar are dependent on the seconds pulse in general mostly counters are used in the digital circuitry to check the particular condition whether it is happening at a particular time or not

B. *Alarm register*

The Alarm register is the functionality similar to the normal register and here the alarm register saves the current alarm values this is used to fix the particular time at which the event must happen

C. *Sound alarm*

Sound alarm compares the current and the alarm time values and makes the sound alarm signal high and the hard ware is generally a comparator[3]

D. *Update register*

Update register is the module and the functionality is simply like a normal register, update register changes its output value for every clock pulse depending on the change of the input signal values

E. *RAM*

RAM [3] consists of the set of registers in which control register and the alarm control register is present and by enabling the required bits in the control registers the required functionality will be obtained

F. *I2C*

I2C [1] is the bidirectional bus here in this paper we are using the I2C slave. SCL clock is generated by the master and the SDA line which is used to transfer both the address and the data the functionality if this bus completely depends on its protocol.

III. DATA TRANSFER WITH I2C

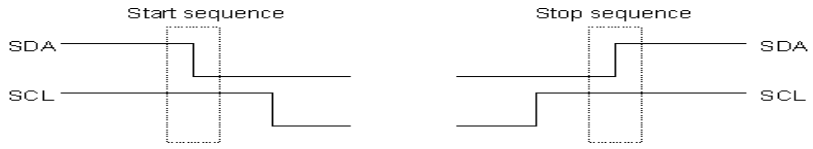

Fig2: start and stop sequence of I2C

When the serial clock is in high position and the data line is from high to low is called as start bit and the same condition with data line low to high is called as stop bit

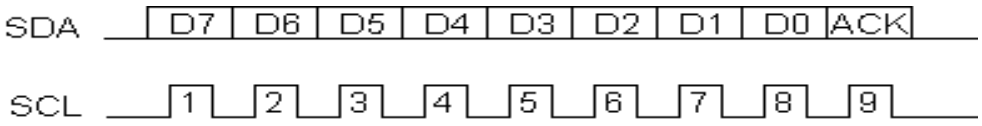

Fig3: SDA and SCL Alignment

Nine clock pulses consists of the 8 bit data and the last pulse is acknowledgement bit which acts as a handshaking signal. The standard clock (SCL) speed for I2C up to 100KHz. Fast mode, which is up to 400KHz and High Speed mode which is up to 3.4MHz. All of our modules are designed to work at up to 100 KHz.

A. *I2C Addressing*

I2C addresses are either 7 bits or 10 bits. The use of 10 bit addresses is rare and is not covered here. All of our modules and the common chips you will use will have 7 bit addresses. This means that you can have up to 128 devices on the I2C bus, since a 7bit number can be from 0 to 127. When sending out the 7 bit address, we still always send 8 bits. The extra bit is used to inform the slave if the master is writing to it or reading from it. If the bit is zero the master is writing to the slave. If the bit is 1 the master is reading from the slave. The 7 bit address is placed in the upper 7 bits of the byte and the Read/Write (R/W) bit is in the LSB (Least Significant Bit)

IV. I2C PROTOCOL

The first thing that will happen is that the master will send out a start sequence. This will alert all the slave devices on the bus that a transaction is starting and they should listen in case it is for them. Next the master will send out the device address. The slave that matches this address will continue with the transaction, any others will ignore the rest of this transaction and wait for the next. Having addressed the slave device the master must now send out the internal location or register number inside the slave that it wishes to write to or read from. This number is obviously dependant on what the slave actually is and how many internal registers it has.

A. Send a start sequence
B. Send the I2C address of the slave with the R/W bit low
C. Send the internal register number you want to write to
D. Send the data byte
E. Optionally, send any further data bytes.
F. Send the stop sequence.

V. WAVEFORMS AND RESULTS

The counter waveform consists of the seconds count pulse and depending on that pulse leap year months and a particular date have been interdependent.

Fig 4: Wave form of the counter

In the below wave form of the update register output changes for every change in the input signal this waveform can be useful in determining the status of the internal signals

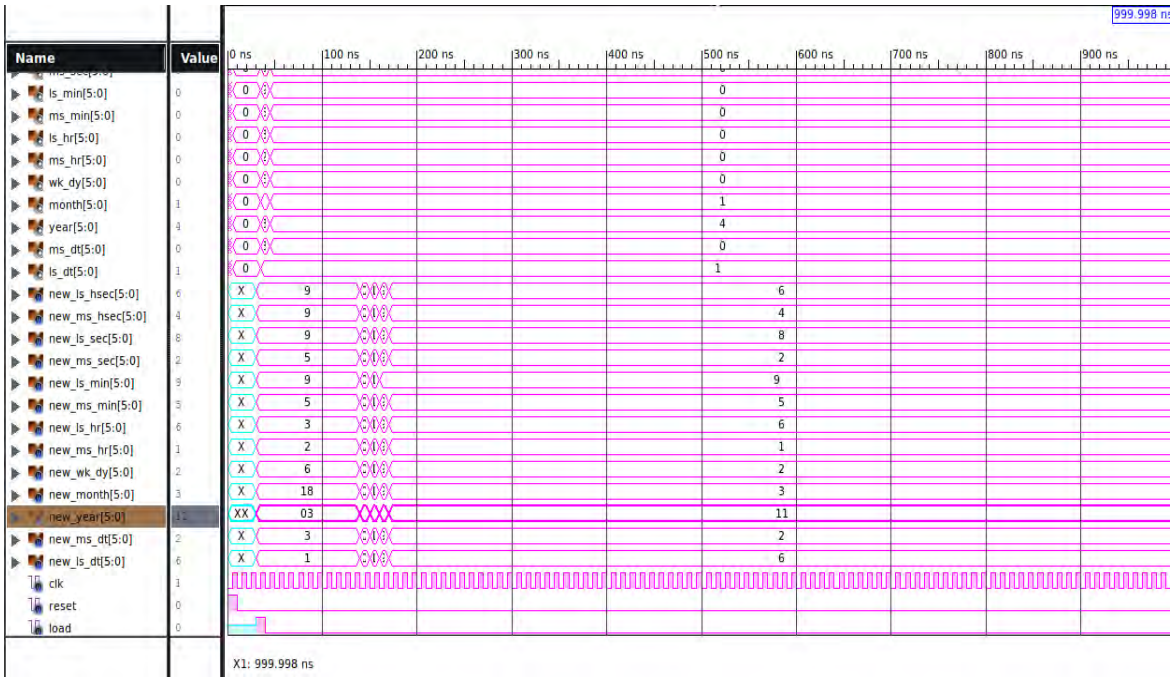

Fig 5: Wave form of update register

In the below wave form of the sound alarm module in which the alarm signal will be high, whenever the current and the alarm register values matches.

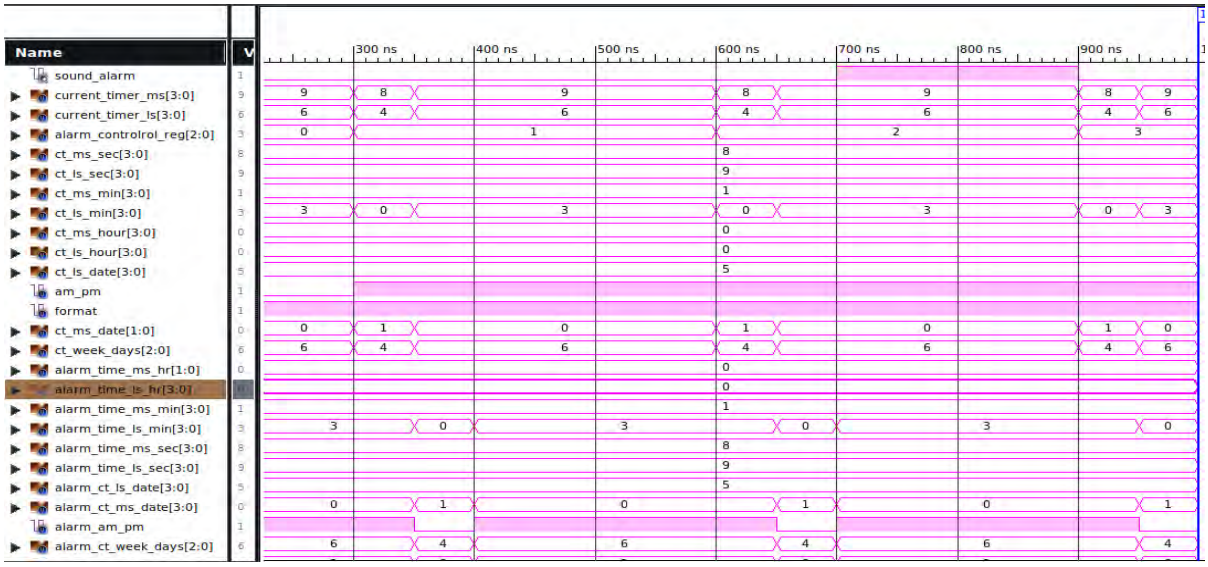

Fig 6: Wave form of sound alarm

In the below wave form of the alarm register is when the reset is in high all the signals will be in the unknown state and when the load value is high input values will truncated on the output values.

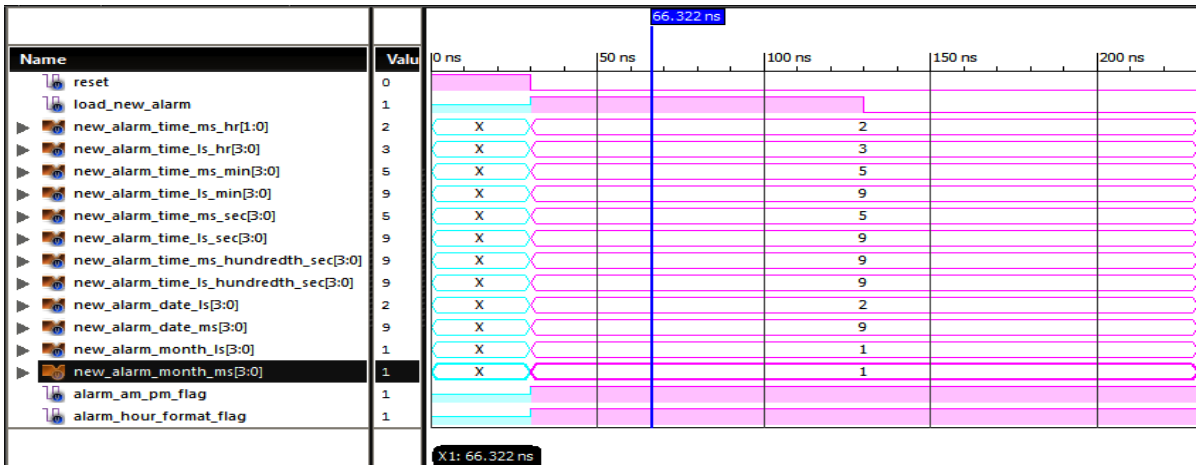

Fig 7: Wave form of alarm register

In the RAM module some write and read signals are in the high impedance because those values must be coming from the other modules this is simply because of module dependency

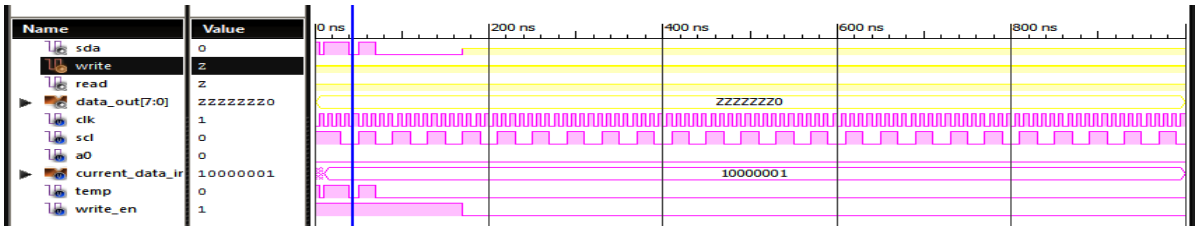

Fig 8: Wave form of the RAM

In this wave form the start and stop bit is clearly noticed at the particular instance and during this start bit the input data is given until the stop bit arrives

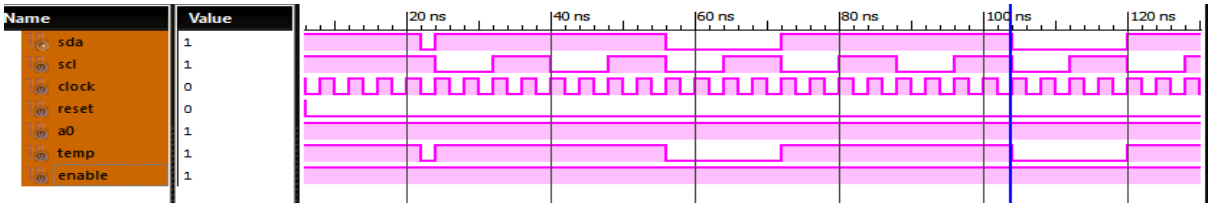

Fig 9: Waveform of the top module

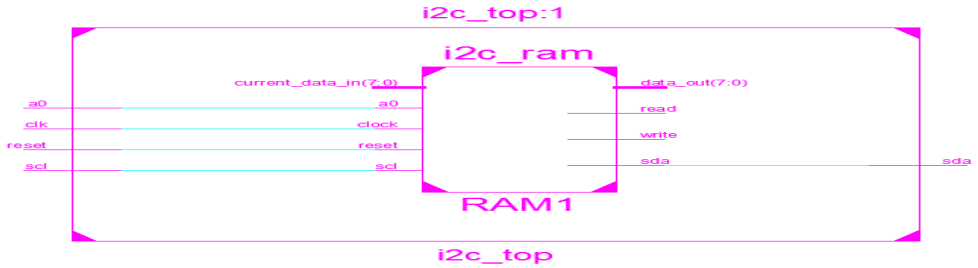

Fig 10: Schematic of the I2C RAM

This is the optimized design of net list obtained from the verilog code using cadence RC extractions and from this the power will be calculated.

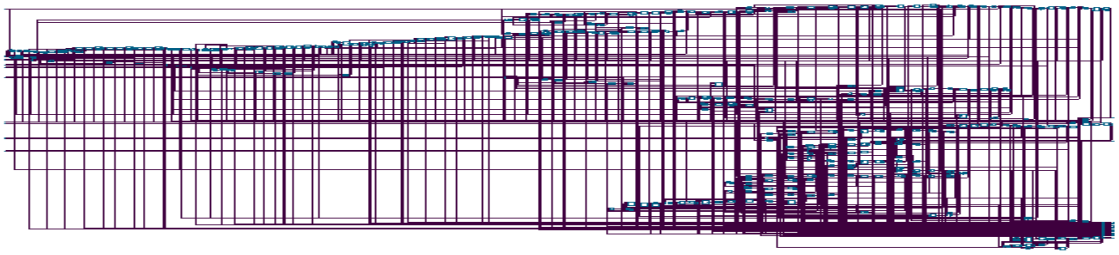

Fig 11: Synthesized diagram of clock/calendar using cadence

The above schematic was obtained from the nclaunch in cadence. Where there is a possibility of checking the syntax errors and the simulation waveforms in the graphical user interface

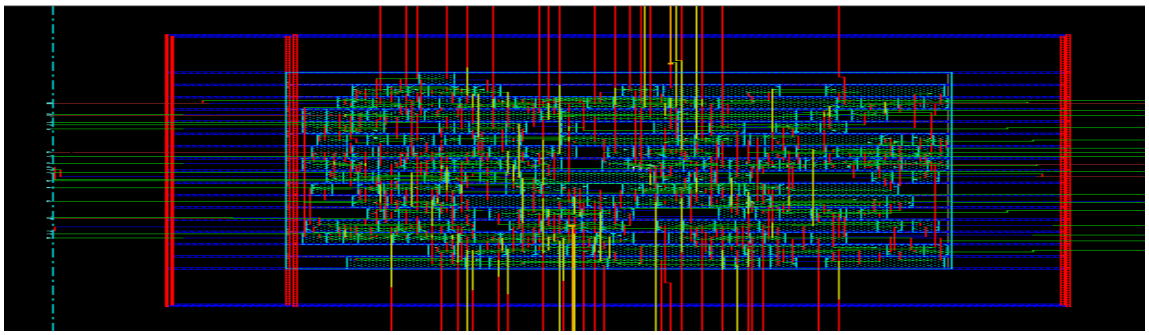

Fig 12: Layout of clock/calendar

The power obtained from the layout is 48624nw which is the best optimized one when compared to the previous models.

VI. Conclusion

The paper can be very useful for various digital circuitry and it reduces the design time of the digital circuits this can be used in various security machines this can be integrated immediately to any device that needs the clock signal of the particular frequency by using cadence 90nm technology the layout has been generated and which is the best and the optimized one obtained from the optimized net-list this can be used in applications such as counters, ALU, and in many of the digital complex circuitry and the power has been calculated which is the optimized one this power has been taken from the layout which is designed in cadence 90nm technology.

REFERENCES

[1] THE I2C BUS SPECIFICATION VERSION 2.1 January, 2000, Philips Semiconductors.
[2] Fred Eady, Networking and Internetworking with Microcontrollers, Elsevier ,2004.
[3] J. Bhasker, A VHDL Synthesis Primer, BS Publications 2nd Edition,
[4] The I2C-bus specification, version 2.1,januaray 2000.
[5] G. Roberts, A. Williams. A LOW-POWER POST-PROCESSED DGPS SYSTEM FOR OCEANOGRAPHIC APPLICATIONS[J]. Electronic Engineering in oceanography. 19-21 July, 1994. pp 18-23
[6] ISL29015 datasheet[OB/DL]. www.intersil.com/data/fin/fn/6522.pdf
[7] Philips standard of analog pcf8583
[8] Intel, Hewlett-Packard, NEC, Dell, Platform Management FRU Information Storage Definition V 1.0, 1999.
[9] Bus Communications Protocol Specification V 1.0, 1999.
[10] YungChin Fang, Garima Kochhar, Randy Deroeck, http://blog.csdn.net/rainlight/archive/2005/12/21/558555.aspx, 2005.8.4.
[11] Haihong Zhuo, Jianwen Yin, Remote Management with the Baseboard Management Controller, Power Solutions, 2004, 26-29.