Indonesian Journal of Electrical Engineering and Computer Science
Vol. 14, No. 2, May 2019, pp. 796~801
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v14.i2.pp796-801

Design and analysis of dedicated Real-time clock for customized microcontroller unit

M.N.A.M. Alias, S.N. Mohyar, M.N. Isa, A. Harun, A.B. Jambek, S.A.Z. Murad
School of Microelectronic Engineering, Universiti Malaysia Perlis, Malaysia

Article Info

Article history:

Received Sep 26, 2018
Revised Nov 27, 2018
Accepted Dec 10, 2018

Keywords:

Advance peripheral bus
Microcontroller
Real-time clock
System-on-chip

ABSTRACT

In this paper, a Real Time Clock (RTC) system for a dedicated microcontroller is proposed to provide the customized microcontroller its own time and date system. The RTC is developed using Verilog Hardware Description Language (HDL) and simulated using Synopsys software. This RTC is developed with standard Advance Peripheral Bus (APB) to be interfacing with the microcontroller through Advanced Microcontroller Bus Architecture (AMBA). This RTC will be used as an on-chip RTC in the microcontroller system to provide precise time and date which can be used for various applications. The basic architecture of RTC, APB standard for interfacing the RTC with AMBA bus, and the result in term of RTL, waveform, and layout will be discussed in this documentation. For this research, the part covered is on the logic part of the RTC that is bus interface, register, frequency divider and counter.

Corresponding Author:

Muhammad Nor Azwan Bin Mohd Alias,
School of Microelectronic Engineering,
Universiti Malaysia Perlis,
02600 Perlis, Malaysia.
Email: azwan_mjg93@yahoo.com

1. INTRODUCTION

A real-time clock is an integrated circuit that contains a timer generated by a crystal oscillator that supplies the time of day and often, with the date. RTC has a small memory on-chip that stores time and date description values. The time and date values are the years, month, date, hours, minutes, and seconds [1]. In this modern era, an electronic device becomes smaller. The smaller size becomes a concern of electronic technology. By using conventional RTC to provide time and date system to a microcontroller will make the device bigger. So, a dedicated RTC for the customized microcontroller is proposed to provide its own time and date system as a system-on-chip feature on the microcontroller [2], [3]. In this project, a dedicated Real Time Clock (RTC) for Customized Microcontroller unit is developed as an Intellectual Property (IP) and will be covered only the logic part of RTC. The development of the dedicated RTC is to provide the customized microcontroller with its own time and date system for various time-based application to be managed precisely without an external RTC to be used with the microcontroller. This documentation continues the proposed architecture of the RTC on the previous work [4] into the ASIC flow for layout generation and some analysis on the layout stage to optimize the layout in term of size and power consumption [5].

2. LITERATURE REVIEW

2.1. Basic Architecture of RTC

From the literature review, there are many designs of RTC nowadays. Every RTC has their own features according to their specification to be used on any application. Even they have different features,

they still have a similar part to make the RTC function correctly such as power management, an oscillation circuit, a communication interface, and memory for time value storage.

The color box in Figure 1 shows the conventional RTC basic architecture. The red box shows the oscillation circuit and prescaler or frequency divider of that RTC. The blue box is the power supply or power management for the RTC. The yellow box will be their register which will store the time and date value generated by the counter. Lastly, green box shows their bus interface for the RTC which will be used to interface the RTC with an external application. All four of this basic architecture is essential for RTC to work correctly.

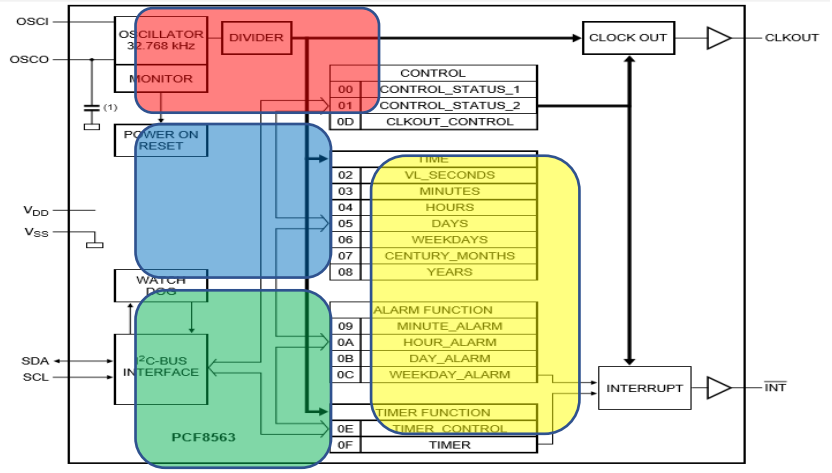


Figure 1. RTC block diagram from NXP semiconductor (PCF8563) [6]

2.2. Oscillation Circuit

An oscillation circuit is an electronic circuit that produces a periodic, oscillating electronic signal, often a square wave. They are widely used in many electronic devices. Common examples of signals generated by oscillators is a clock signal that regulates computers and quartz clocks. Some of the factors that affect the frequency stability of an oscillator generally include the variations in temperature, variations in the load as well as changes to its DC power supply voltage and many other factors. The frequency stability of the output signal can be improved by the proper selection of the components used for the resonant feedback circuit including the amplifier. The frequency stability is needed in RTC because it plays an important role. The clock generated will base on the frequency and if the frequency is not stable, it will affect the counting thus affect the accuracy of the RTC.

To obtain a very high level of oscillator stability, a Quartz Crystal is generally used as the frequency determining the device to produce another type of oscillator circuit known generally as a Quartz Crystal Oscillator (XO). There are many different types of crystal substances which can be used as oscillators with the most important of these for electronic circuits being the quartz minerals because of their greater mechanical strength. The physical size and thickness of a piece of quartz crystal are tightly controlled since it affects the final or fundamental frequency of oscillations. Then once cut and shaped, the crystal cannot be used at any other frequency. In other words, its size and shape determine its fundamental oscillation frequency [7].

Also, in the crystal oscillator, there are many types of the crystal oscillator with a different configuration for different specification such as Colpitts Crystal Oscillator, Pierce Oscillator, and CMOS Crystal Oscillator. As for their oscillation frequency, all RTC use the 32.768kHz crystal oscillator due to the stability of the crystal [8]-[10].

2.3. Bus Interface

For conventional RTC, there are two types of interface technique used which is I²C (Inter-Integrated Circuit protocol) and SPI (Serial Peripheral Interface). These two interfaces have their own advantage depending on their application. Both protocols are well-suited for communications between integrated circuits, or with onboard peripherals [11].

I²C is developed by Philips (now known as NXP Semiconductor). It is a simple bidirectional 2-wire bus for efficient inter-IC control. With this bus interface, only two bus lines are required that is a serial data

line (SDA) & a serial clock line (SCL). Serial, 8-bit oriented, bidirectional data transfers can be made at up to 100 kbit/s in the Standard-mode, up to 400 kbit/s in the Fast-mode, up to 1 Mbit/s in the Fast-mode Plus, or up to 3.4 Mbit/s in the High-speed mode. The Ultra-Fast-mode is a uni-directional mode with data transfers of up to 5 Mbit/s [12].

SPI is developed by Motorola. This interface provides full-duplex synchronous serial communication between master & slave devices. It is commonly used for communication with the flash memory, sensors, real-time clock (RTCs), analog to digital converter and many more [13]

2.4. Advanced Peripheral Bus (APB)

The propose RTC [4] is a dedicated RTC for a microcontroller and will be as an onboard RTC. The microcontroller will use an Advanced Microcontroller Bus Architecture (AMBA) bus protocol as their bus interface. The proposed RTC needs to follow the AMBA bus protocol to be interfaced with the microcontroller bus since this is the standard protocol for an onboard communication. To easily interface the IP with the microcontroller, the RTC interface needs to match with Advanced Peripheral Bus (APB) specification. APB is part of the AMBA hierarchy of buses. It is optimized to reduce the complexity of interfacing and for minimalize power consumption. The AMBA and APB should be used to interface to any peripherals which not require high performance of bus interface and low bandwidth [14]-[19]

Since this IP will be integrated with the microcontroller that using AMBA Bus Architecture, the RTC design then need to follow the APB Specification as shown in Figure 2.

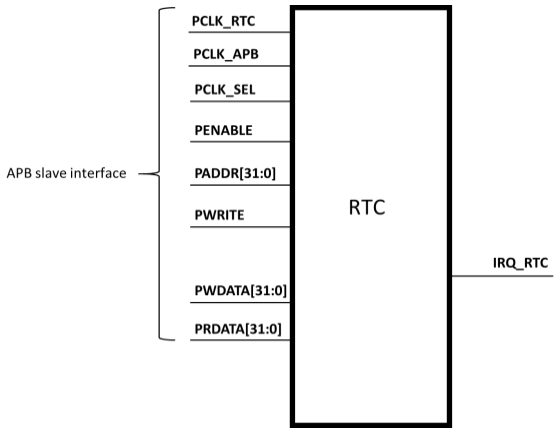


Figure 2. APB specification interface

2.5. Memory/Register

A small memory on this microchip stores system description or setup values including current time values stored by the real-time clock. The time values are for the seconds, minutes, hours, date, month, and years.

3. METHODOLOGY

From the proposed architecture in the previous work[2], the work continues from FPGA prototyping into physical implementation (layout) through Application Specific Integrated Circuit (ASIC) design flow [20].

3.1. ASIC Design Flow

Figure 3 shows the ASIC design flow which starts with a logic design until physical implementation (layout) for fabrication. This work will be using 0.18µm Silltera Technology library and process through Synopsys software; VCS, Design Compiler, IC Compiler.

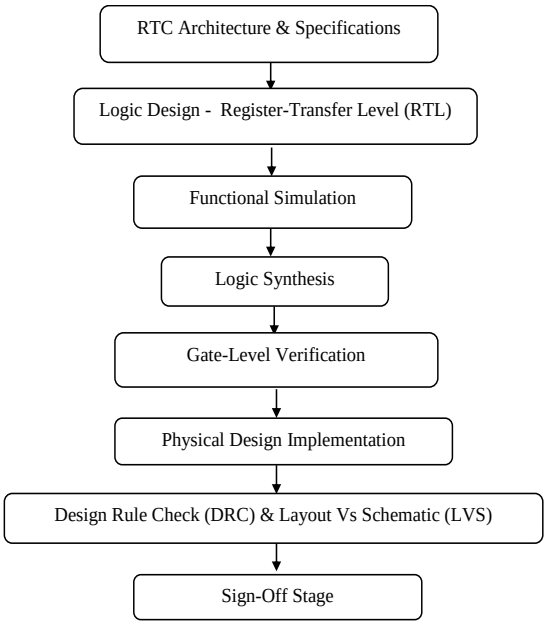


Figure 3. ASIC flow of RTC design

4. RESULTS AND DISCUSSION

4.1. Top Level RTL

Figure 4 shown the Top-level RTL of the proposed RTC generated using VCS software which consists of five sub-module; a frequency divider, function controller, rtc, alarm and outdata module. The function of each module is as below:

a. Frequency Divider – To generated ±1kHz clock source for rtc module from the 32.768kHz input clock.
b. Function Controller – To control the function bit on the design.
c. Rtc Module – Counter module which will count and store the time and date description value.
d. Alarm Module – Module which will store the alarm set value and compare with the real-time value to give interrupt signal when match.
e. Outdata Module – Store concatenation value of the time and date value in 32-bit bus.

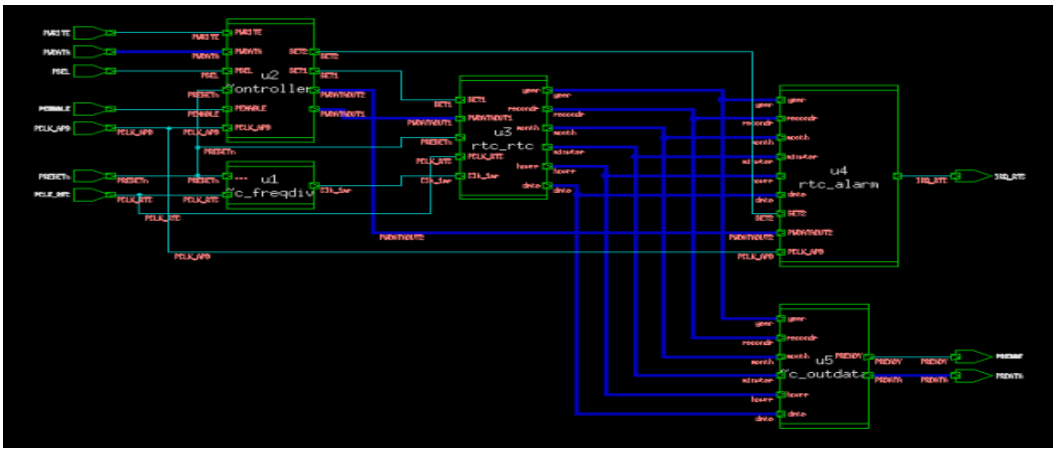
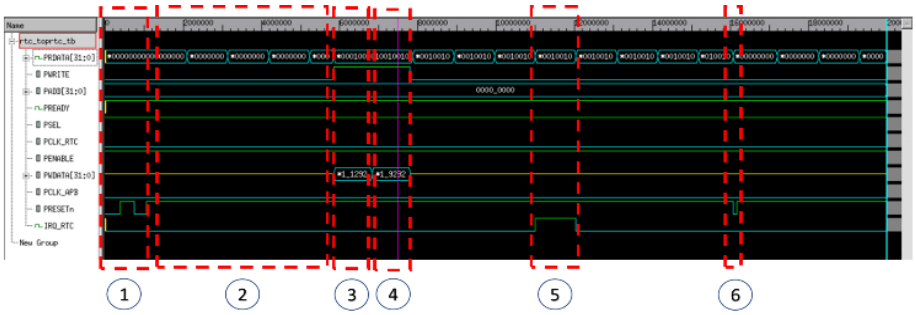


Figure 4. Top level RTL of the RTC

4.2. Simulation Waveform

Figure 5 shows the functional output waveform of the proposed Top level RTC IP generated using VCS and Design Compiler software. The whole waveform shows all the condition as stated below the waveform. Initialize condition is to initialize the IP by performing 2 cycles of a reset signal on the IP.

The first cycle is to reset the frequency divider module to initialize the value of register inside the module before it can start dividing the input clock source into desired clock source for the counter. For the second cycle of reset, it will reset all condition to its initial condition. The counting condition will start immediately after the reset is 1(active low reset). The counter will start counting in the rtc module with a resolution of ±1millisecond. The manual set and alarm set will be triggered by a certain condition stated in the previous work [4]. The alarm will be triggered when the real-time value is matched with the value of the alarm set. It will send an interrupt signal to indicate the set value is matched with the real-time value. The reset condition will be triggered by a low reset signal. This will reset all the value in the rtc and alarm module to its initial condition.



Number	1	2	3	4	5	6
Condition	Initialize/reset	Counting	Manual Set	Alarm Set	Alarm Triggered	Reset

Figure 5. Simulation waveform of the RTC

4.3. Generated Layout

The layout is generated using 0.18µm Silttera Technology on IC Compiler software (Synopsys). The optimization and analysis are done on generated the layout in term of power and size. Figure 6 shows the first layout generated by using width/height control type in the floorplan stage while Figure 7 is the optimize layout by using 0.9 core utilization on the floorplanning stage. By using the core utilization control type for the floorplan, the size is reduced by 70%. This is because this control type will utilize 90% of the core size for the standard cell while width/height control type will ignore the utilization of the core and will create the core size as the defined value. In term of power, the power consumption of the optimized layout is reduced by creating a smaller power/ground ring. The power consumption is also reduced by the smaller core itself since a smaller core will contribute to a shorter routing. The power consumption difference between first and optimize layout is reduced by 9%. Table 1 shows the comparison of the first layout and optimizes layout generated in term of size and power consumption

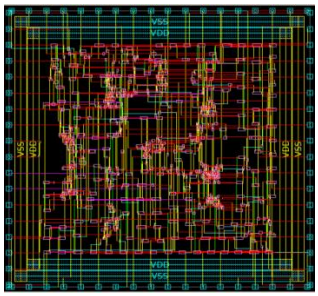


Figure 6. First layout of the proposed RTC

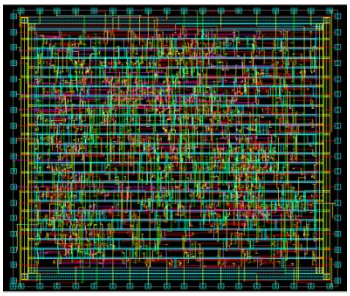


Figure 7. Optimized layout of the proposed RTC

Table 1. Comparison of a First and Optimized Layout of Proposed RTC

Layout Specification	First layout	Optimized layout
Size (width x height)	500µm x 500µm	150µm x 150µm
Dynamic power (µW)	137.35	125.02
Leakage power (nW)	512.26	518.44
Total power consumption(mW)	0.1379	0.1255

5. CONCLUSION

In a conclusion, the proposed design of the RTC in the previous work has been put through the ASIC design flow to generate the layout. The generated layout then is optimized in term of size and power consumption. This work will be continued to the sign off stage before being sent out for fabrication. This IP will be integrated into the microcontroller alongside with the other IP to be fabricated. The main contribution of this RTC development is to provide the customized microcontroller with its own time and date system to manage various time-based application without external RTC being used with the microcontroller.

ACKNOWLEDGEMENTS

This project is sponsored by Talent Corp. Malaysia in collaboration with Intel Malaysia and CEDEC USM for 3D (Demand, Driven, and Develop) program. The author also would like to acknowledge the support from the Tabung Penyelidikan Jangka Pendek under a grant number of 9009-00025 from Universiti Malaysia Perlis (UniMAP).

REFERENCES

[1] "Real-Time Clocks (RTC) ICs - Maxim", Maximintegrated.com, 2016. [Online]. Available:https://www.maximintegrated.com/en/products/digital/real-time-clocks.html. [Accessed: 27- Feb- 2018].
[2] Javier Jalle, Mikel Fernandez, Jaume Abella, Jan Andersson, Mathieu Patte, Luca Fossati Marco Zulianello, Francisco J. Cazorla,"Contention-Aware Performance Monitoring Counter Support for Real-Time MPSoCs", 2016 11th IEEE Symposium on Industrial Embedded Systems (SIES), 23-25 May 2016, Krakow, Poland.
[3] Nileshkumar Pandey and Doina Bein, Web application for social networking using RTC,2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC), 8-10 Jan. 2018, Las Vegas, NV, USA
[4] Mohd Alias, M.N.A., Mohyar, S.N. "Architectural design proposal for a real-time clock for wireless microcontroller unit" (2017) EPJ Web of Conferences, 162, art. no. 01072
[5] David Ruffieux, Franz Pengg, Nicola Scolari, Frédéric Giroud, Daniel Severac, Thanh Le, Silvio Dalla Piazza, Olivier Aubry, "11.5 A 3.2×1.5×0.8mm3 240nA 1.25-to-5.5V 32kHz-DTCXO RTC module with an overall accuracy of µ1ppm and an all-digital 0.1ppm compensation-resolution scheme at 1Hz", 2016 IEEE International Solid-State Circuits Conference (ISSCC), 31 Jan.-4 Feb. 2016, San Francisco, CA, USA
[6] NXP semiconductor, "Real-time clock/calendar," *PCF8563*, Rev.11, datasheet, Oct. 2016
[7] "Quartz Crystal Oscillator and Quartz Crystals", Basic Electronics Tutorials, 2017. [Online]. Available: http://www.electronics- tutorials.ws/oscillator/crystal.html. [Accessed: 12- Nov- 2017]
[8] Aatmesh Shrivastava, Divya Akella Kamakshi, and Benton H. Calhoun, "1.5 nW, 32.768 kHz XTAL Oscillator Operational From a 0.3 V Supply", IEEE Journal of Solid-State Circuits (JSCC), Vol. 51, No. 3, March 2016.
[9] Ju-Hyun Park, Sung Han Do, Sung Jin Kim, Ho-Cheol Ryu, and Kang-Yoon Lee, "A Design of 4.19MHz Real Time Clock Generator with Triple mode for Fast settling, Current Reduction and Low Noise in 0.18um CMOS ", 2016 International Conference on Electronics, Information, and Communications (ICEIC), 27-30 Jan. 2016, Da Nang, Vietnam.
[10] Pyoungwon Park, David Ruffieux, and Kofi A. A. Makinwa, "A Thermistor-Based Temperature Sensor for a Real-Time Clock With 2 ppm Frequency Stability ", IEEE Journal of Solid-State Circuits (JSCC), Vol. 50, No. 7, 2015.
[11] "Introduction to I²C and SPI protocols – Byte Paradigm – Speed up embedded system verification", Byteparadigm.com, 2017. [Online]. Available: http://www.byteparadigm.com/applications/introduction-to-i2c-and-spi-protocols/. [Accessed: 12- Nov- 2017].
[12] NXP Semiconductor, "I2C-bus specification and user manual," *UM10204*, Rev. 6 ,User Manual, Apr 2014.
[13] "SPI Interface - Tutorial", Corelis.com, 2017. [Online]. Available:http://www.corelis.com/education/SPI_Tutorial.htm. [Accessed: 12- Nov- 2017]
[14] "AMBA specification 2.0," ed: ARM Ltd., 1999.
[15] "Cortex-M System Design Kit Technical Reference Manual," ed: ARM Ltd., 2011
[16] Abhijeet Paunikar, Rohan Gavankar, Nachiket Umarikar, K.Sivasankaran, "Design and Implementation of Area Efficient, Low Power AMBA-APB Bridge for SoC", 2014 International Conference on Green Computing Communication and Electrical Engineering (ICGCCEE), 6-8 March 2014, Coimbatore, India.
[17] Kiran Rawat, Kanika Sahni, Sujata Pandey, "RTL implementation for AMBA ASB APB protocol at system on chip level", 2015 2nd International Conference on Signal Processing and Integrated Networks (SPIN), 19-20 Feb. 2015, Noida, India.
[18] Kiran Rawat, Kanika Sahni, Sujata Pandey, Jyoti Rawat and SudhanshuTripathi, "Implementation of AMBA APB bridge with efficient deployment of system resources", 2015 International Conference on Computer, Communication and Control (IC4), 10-12 Sept. 2015, Indore, India
[19] Anshu Gaur, Piyush Sharma and Shiv Pratap Pandey, "HDL and timing analysis of AMBA AHB on FPGA platform", 2017 Recent Developments in Control, Automation & Power Engineering (RDCAPE), 26-27 Oct. 2017, Noida, India.
[20] Ching-Wen Hsu and Kuei-Ann Wen, "Monolithic Integration of Digital MEMS Thermometer and temperature compensated RTC on 1P6M ASIC compatible CMOS MEMS process", 2018 IEEE International Conference on Semiconductor Electronics (ICSE), 15-17 Aug. 2018, Kuala Lumpur, Malaysia.