Integration of Real-Time Clock Unit into Reconfigurable 8051 IP-core

Murat ÇAKIRO LU[1], Ahmet Turan OZCERIT[1]

[1]Department of Computer Engineering, Faculty of Technology, SAKARYA University, Turkey
Email:{muratc, aozcerit}@sakarya.edu.tr

ABSTRACT

This paper presents an implementation of Real-Time Clock (RTC) unit integrated within the 8051 microcontroller by means of a hardware description language. The verification of the RTC has been done by using a commercial Field Programmable Gate Array (FPGA) development environment deployed with a simulator and a synthesizer. Throughout the modelling procedures, the authentication of the 8051 architecture is preserved as much as possible.

INTRODUCTION

Modern approach in electronic system design is called system-on chip (SoC) [1]. In this technology, the units forming the system are described in a hardware description language and downloaded to the target device after synthesizing procedures. Throughout whole processes of this technology, a couple of commercially available software-based development systems are used to implement the system. Such systems can be implemented by FPGA devices since their huge capacity and great deal of flexibility [2,3]. The FPGA devices nowadays have proved to be popular method of implementation of such systems [4].

Although the 8051 microcontroller has been a standard for the industrial applications it needs some additional hardware units and special routines for real-time applications. It is clearly appeared that, the system designed comprising a 8051 microcontroller should be implemented by the help of a reconfigurable architecture to keep up with the technology mentioned above [2] and to be readily adapted to a basic configuration for real-time applications.

FUNCTIONAL BLOCK CONFIGURATIONS

In Figure 1, the block diagram of the reconfigured 8051 comprising of control unit, ALU, timer/counter, serial communication units, program and data memory can be seen. The bus control unit has been tailored to facilitate up to 256 byte data memory and up to 64Kbyte program memory interface.
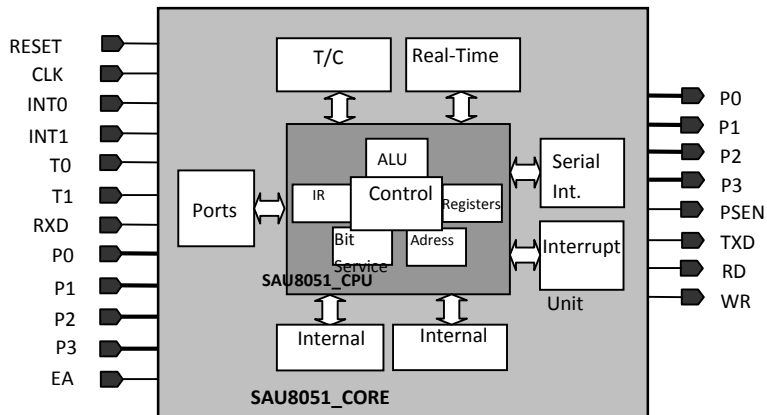
Figure 1 The block diagram of the core designed.

The 8051 designed has following standard features:

- 32 I/O pins
- Up to 256 Bytes internal RAM
- Up to 64 Kbyte internal ROM
- Up to 64 Kbyte external ROM and RAM
- 2 timer/counter
- 1 serial communication interface
- 2 external interrupt source
- RTC unit

There are few constant parameters that specify the standards of the 8051. The VHDL source code has been designed in such a way that these constants can be changed at will giving an opportunity to the user to specify 8051 core configuration. Thus, data and program memory sizes, command set, number of interrupt sources, number of timer/counter and serial communication units can be determined as requested.

PROPERTIES OF THE REAL-TIME CLOCK CIRCUIT

The real-time clock circuit in the Timer/Counter block consist of a 32-bit Master Counter (MC) register, five 8-bit registers (SEC, MIN, HOUR, DAY and MON), a 16-bit "YEAR" register, and an 8-bit "RTCON" register to control the real-time counting. The MC register counts by 1-second intervals in respect to the selected crystal values. The SEC, MIN, HOUR, DAY, MON and YEAR registers keep the information about second, minute, hour, day, month and year respectively. Register organization in Special Function Registers (SFR) is shown in Figure 2.

MCLL	MCLH	MCHL	MCHH				
RTCON	SEC	MIN	HOUR	DAY	MON	YEARL	YEARH

Figure 2

RTC register in Special Function Registers (SFR)

As shown in Figure 3, RTCON is a SFR register and has five control bits. The Bit0, which is also called EN, enables/disables Real-Time counter. When EN is high the RT-Counter starts to run and it stops when EN is low. The second control bit is the CLR bit is used to set all RT-Counter registers to their default levels. The third, fourth and fifth bits are reserved for future use.

OSC2	OSC1	OSC0	--	--	--	CLR	EN

Figure 2 RTCON register

The Bit5, Bit6, and Bit7 of the RTCON register can be used to determine clock frequency for the RT-Counter. The possible clock frequencies are shown in Table 1. The OSC0, OSC1, and OSC2 bits should be set by the user according to crystal frequency to be used.

Table 1 Crystal frequency configuration bits

OSC2	OSC1	OSC0	Crystal Frequency (Mhz)
0	0	0	6
0	0	1	10
0	1	0	12
0	1	1	20
1	0	0	24
1	0	1	48
1	1	0	-
1	1	1	-

THE OPERATION OF THE REAL-TIME CLOCK MODULE

According to the values of the Bit5, Bit6, and Bit7 of the RTCON SFR register, whatever crystal frequencies, which are given in Table 1, have been selected, 1µsec-pulse is produced by means of clock divider unit. 32-bit register MC counts those pulses produced by the divider unit and overflows when the count value reaches 106 counts. Thereby the "SEC register" is updated, in other words, it is incremented by one. This

procedure imitates the real calendar working cycle with taking into consideration leap-years. In Figure 4, the functional architecture of the RTC circuit can be seen. The basic flow chart of the RTC mechanism is shown in Figure 5 below.
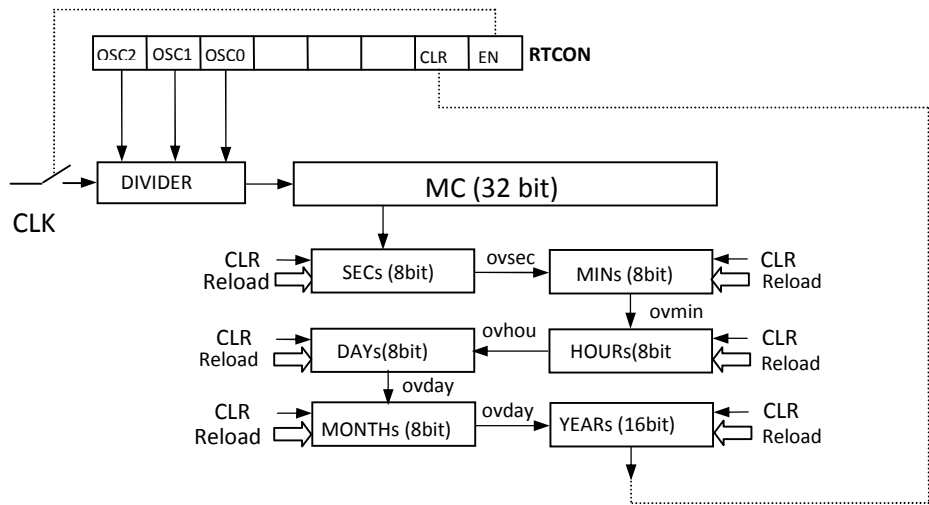

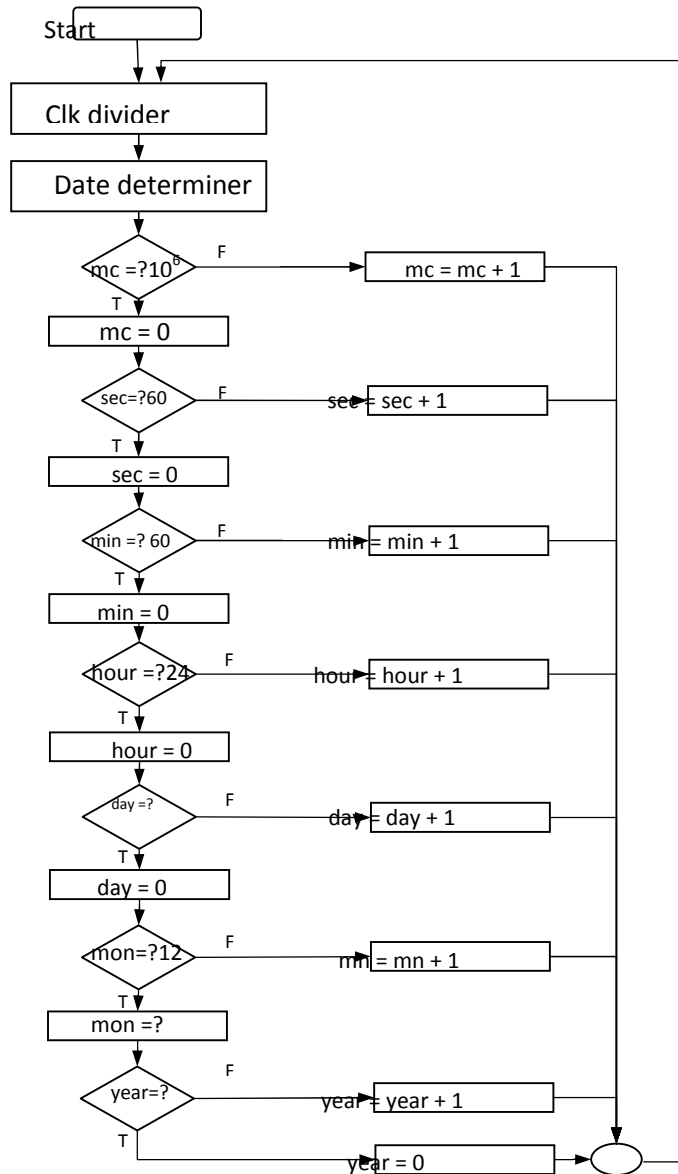Figure 4 The architecture of RTC circuit

```
                    ┌──────────┐
                    │  Start   │
                    └────┬─────┘
                         │    ◄──────────────────────────────────┐
                    ┌────▼─────────┐                              │
                    │  Clk divider │                              │
                    └────┬─────────┘                              │
                    ┌────▼──────────────┐                         │
                    │  Date determiner  │                         │
                    └────┬──────────────┘                         │
                       ◄─┴─►    F      ┌──────────────┐           │
                    <mc =?10⁶>─────────►│ mc = mc + 1 │───────────┤
                         │ T            └──────────────┘          │
                    ┌────▼─────┐                                  │
                    │  mc = 0  │                                  │
                    └────┬─────┘                                  │
                    <sec=?60>──F──────►│ sec = sec + 1 │──────────┤
                         │ T           └───────────────┘          │
                    ┌────▼─────┐                                  │
                    │  sec = 0 │                                  │
                    └────┬─────┘                                  │
                    <min =? 60>──F─────►│ min = min + 1 │─────────┤
                         │ T            └───────────────┘         │
                    ┌────▼─────┐                                  │
                    │  min = 0 │                                  │
                    └────┬─────┘                                  │
                    <hour =?24>──F─────►│ hour = hour + 1 │───────┤
                         │ T            └─────────────────┘       │
                    ┌────▼─────┐                                  │
                    │ hour = 0 │                                  │
                    └────┬─────┘                                  │
                    <day =?>──F────────►│ day = day + 1 │─────────┤
                         │ T            └───────────────┘         │
                    ┌────▼─────┐                                  │
                    │  day = 0 │                                  │
                    └────┬─────┘                                  │
                    <mon=?12>──F───────►│ mn = mn + 1 │───────────┤
                         │ T            └─────────────┘           │
                    ┌────▼─────┐                                  │
                    │ mon =?   │                                  │
                    └────┬─────┘                                  │
                    <year=?>──F────────►│ year = year + 1 │───────┤
                         │ T            └─────────────────┘       │
                         └──────────►│ year = 0 │──►( )───────────┘
```

Figure 5 Simplified flow chart of the RTC unit

SIMULATION RESULTS

In Figure 6, the simulation results of the RTC function can be seen with only critical and significant section of the waveform. As shown in Figure 6, crystal frequency

is selected as 10MHz and RTCON register is set in order to enable the RTC unit (RTCON=00100001). To easily observe the changes on signals, second, minute, hour, day, month, and year registers are loaded with on the verge of limit values. The SEC register is incremented by one when the MC register reaches to 999999 (1µsec x 106) value, which means one second is elapsed. As a result, the successive registers, which are cascaded each other, are updated accordingly.
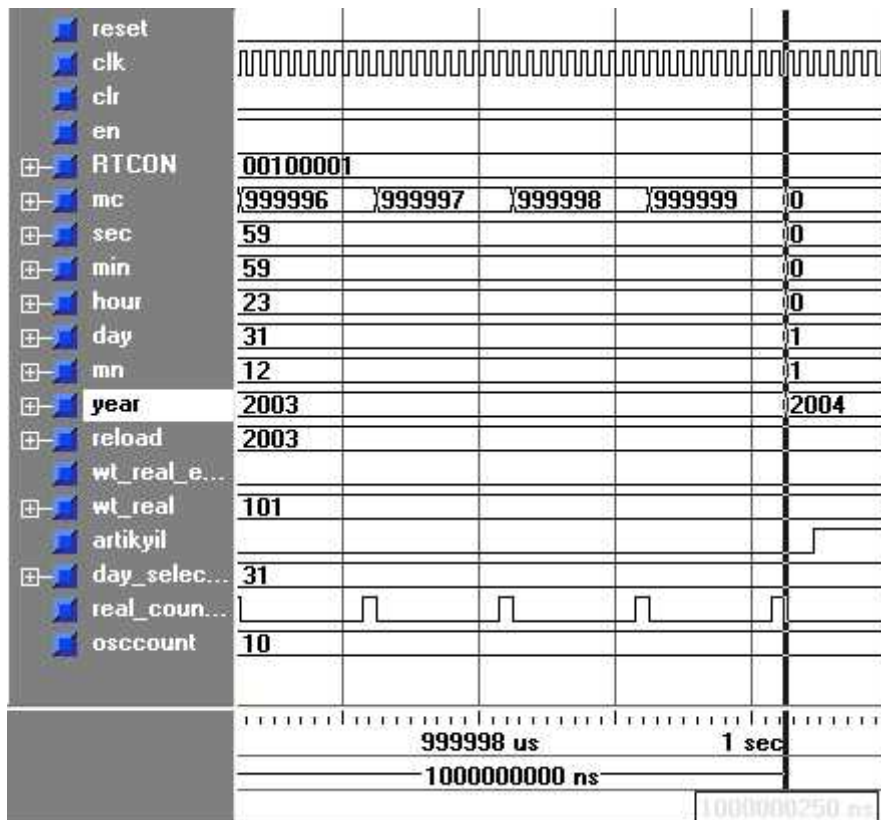

Figure 6 The simulation of the Real-Time Clock Circuit.

CONCLUSION

In this work, we have investigated the on-chip RTC circuits for the 8051 microcontrollers to be used in embedded system applications. The initial aim was to incorporate the RTC unit into the 8051 by means of reconfiguration of 8051 IP-core coded in VHDL. The whole 8051 IP-core has been synthesized and verified under FPGA development environment supplied by Xilinx ISE and Mentor Graphics FPGA Advantage 5.4 Pro incorporating ModelSim 5.6f and Leonardo Spectrum 2002e. The target device to be used for this application is Virtex-II from Xilinx.

The on-chip RTC circuit saves space and simplifies interface between the microcontroller and itself as well as preparing a well defined and superior environment for embedded programmers. However, it still requires some additional arrangements such as independent crystal frequency, battery backed memory and external interface for a battery.

ACKNOWLEDGEMENTS

REFERENCES

[1]   Borgatti, M., Lertora, F., Foret, B., Cali, L. : A Reconfigurable System Featuring Dynamically Extensible Embedded   Microprocessor, FPGA and Customizable I/O, IEEE Custom Integrated Circuits Conference, (2002) 13-16

[2]   Bezarra, E.A, Gough, M.P,: A Guide to Migrating from microprocessor to FPGA Coping with the Support Tool   Limitations, ELSEVIER Microprocessor and Microsystems 23, (1999) 561-572

[3]   Salcic, Z.: PROTOS- A Microcontroller/FPGA-based Prototyping System for Embedded Applications, ELSEVIER Microprocessor and Microsystems 21, (1997)   249–256

[4]   Zhang, X., Ng, W., Kam, : A Review of high-level synthesis for dynamically reconfigurable FPGAs. ELSEVIER Microprocessor and Microsystems 21, (2000) 199–211