LOW POWER REDUCED INSTRUCTION SET ARCHITECTURE USING CLOCK GATING TECHNIQUE

Dr.M.Kamaraju[1] and G.Chinavenkateswararao[2]

[1]Professor & Head, Dept. of ECE, Gudlavalleru Engineering College, Gudlavalleru – India
[2]M.Tech DECS group, Dept. of ECE, Gudlavalleru Engineering College, Gudlavalleru – India

ABSTRACT

Today, all the portable device's in electronics needs to be realized with low power architectures because of power consumption is a main consideration along with other performance parameters. Low power consumption helps to reduce heat dissipation, increases battery life and also reliability. In this paper a 16-bit Reduced Instruction set Architecture (RISA) presented. This architecture can handle multiple interrupts and performing serial communication effectively. It can supported RISC (Reduced Instruction Set Computer) concepts. A popular technique of Clock gating is applied to the proposed architecture and then reduces the power. This entire architecture captured using VerilogHDL and implemented on FPGA using Xilinx tools.

KEYWORDS

Low power, RISA, RISC, FPGA, Interrupt, Clock- gating.

1. INTRODUCTION

There are several reasons for emphasizing low power dissipation in modern processor [2] designs. Some of the device performance related issues, while others may be manufacturing issues. The most of embedded systems today operated with battery, the core of the embedded systems may be the processor or controller. If the core of the system is consuming more power, then life of battery also decreases. Particular in low power deign, the core of the system design to consume low power. It leads to increases battery life of the embedded system. Another performance related issue is that, in many cases, reducing the switching current in the processor increases the reliability of the device. A Reduced Instruction Set Architecture (RISA) is a processor that has been embedded into a device. Performance wise, an embedded processor [3] can outperform a microcontroller, but does not have as much performance as a general-purpose microprocessor.

In this paper describes the architecture of RISA using clock gating particularly designed for collection of data and transfer the packetized data [9] to the terminal station based on more number of interrupts which are coming from external devices and then applying clock gating (anding). The technique mainly concentrates on reducing power and also establishing serial communication between two systems.

DOI : 10.5121/vlsic.2013.4503

In the earlier designs, the designers thought that the clock signal is a clean signal and should not be disabled. But it is one of the major sources for power dissipation because clock signal is feed to most of circuit blocks in the architecture. It leads to unnecessary dynamic power consumption. In this architecture, all the blocks are not operated at the same and a chance to reduce the power consumption of unused block.

This architecture allows the number of interrupt signals coming from various internal or external devices; in which the process of serving request provided by reduced instruction set central processing unit (RISCPU). It can handle a more number of interrupts based on interrupt handler mechanism. So in this paper describes, separate interrupt controller which is interfaced to the processor and also separate transmitter and receiver modules are interfaced. This increases the complexity of design and speed of the processor.

The following sections will provide a brief overview of architecture of the RISA with clock gating and explanation about the implementation. Then brief description about the RISC, Interrupt controller, Transmitter and Receiver sections of the RISA with clock gating are given. The block diagram of proposed architecture as described in first section. Later the individual modules in architecture are explained. Finally few notes on simulation results.

2. PROPOSED ARCHITECTURE

2.1 Clock Gating Technique

Clock gating technique [1] is used in the proposed RISA architecture. Clock power is important component of overall dynamic power consumption. One way to reduce clock power using clock gating (anding) [4], Which dynamically disables the clock signals in unused modules of the circuit as shown in figure 1. This avoids the unnecessary power dissipation caused by charging and discharging the clock signal at unused gates.

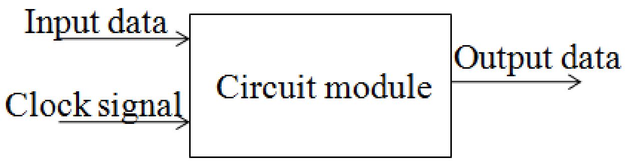


Figure 1. Circuit module without clock gating technique

Clock-gating [5], [6] is a technique where the clock signal is prevented from reaching the various modules of the processor. The absence of the clock signal prevents any register and or flip-flop from changing there value. Gating is achieved by 'ANDing' the clock signal with a clock gated control signal, Whenever any module of circuit that needs to be gated clock is active otherwise low as shown in figure 2. As a result of this, the input to any combinational logic circuit remains unchanged, and thus no switching activity takes place in those circuits.

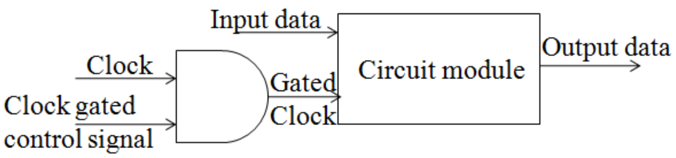


Figure 2. Circuit module with clock gating technique

In this Architecture, mostly concentrated on the ALU, Port controller, Interrupt Controller, Transmitter and Receiver modules for clock gating technique. The reason for this is that these are the biggest modules in our architecture in terms of number of logic gates. The Control Unit is responsible for generating the clock gating signal based on the current instruction.

2.2 RISA Block Diagram

The proposed RISA Architecture mainly consists of RISCPU, Interrupt controller, Port controller, Program flow controller, Transmitter and Receiver and its block diagram is shown in the Figure 3. These blocks are connected by 16-bit internal buses.

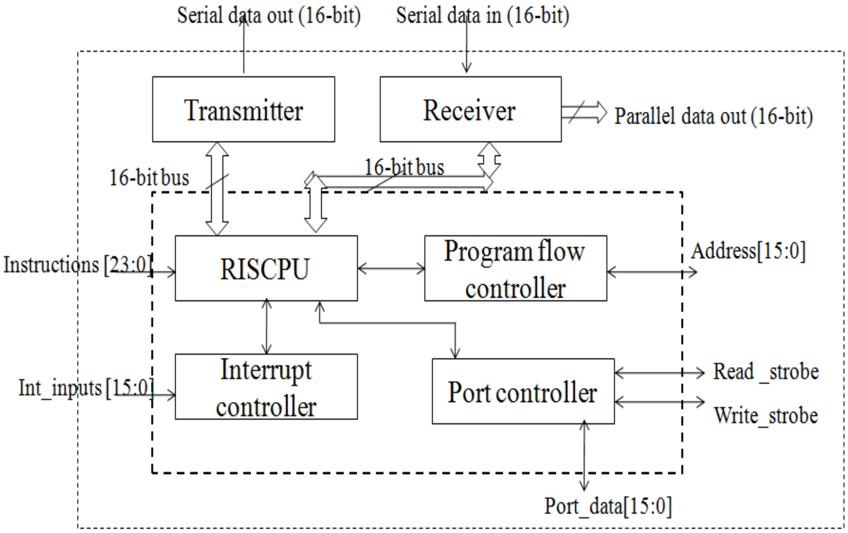


Figure 3. Block diagram of RISA

The instruction length of the RISA is 24-bit wide. The RISCPU has three flags namely carry, zero and interrupt flags. The arithmetic and logical instructions results are affected by zero and carry flags and these are useful for determine the execution flow of jump and branch instructions. Central processing unit (CPU) verifies the interrupt flag after completion of every ALU instruction to know whether the interrupt signal is obtained or not. Along with this a stack is used to store up to four (16-bit) addresses during interrupt and branch related instructions.

Interrupt controller handles single write port and multi read port. Write and read operations are performing during the negative edge and positive edge of the clock signal. Port controllers take care of the write and read operation. A 16-bit address value provided on the “port” bus together with a ‘read and write strobe’ signals indicated the accessed port. The port address is used to provide an absolute value in a program or identified indirectly as the contents of any of the eight registers. The controlling of interrupt controller presenting in a RISA using some specific interrupt instructions.

The data transfer between to and from the architecture using a special function register (16-bit) is provided in the RISA. By using a special function register (SFR) are easy to interfacing between the transmitter and receiver modules of the architecture. It is also acts as a control word register. Some application specific instructions are particularly used for the data transfer and received from sensor locations. Several Instruction Set Architecture for various applications have been proposed in [10].

3. INTERRUPT CONTROLLER, TRANSMITTER AND RECEIVER SECTIONS OF RISA

3.1. Interrupt Controller

The Interrupt controller [7], [8] is a device commonly found in a computer systems, which deals with the interrupts are generated by number of sources or peripherals, both internal and external. It is serves to combine the interrupt requests on a priority (interrupt handling mechanism) basis for individual service provided by the RISCPU. It contains Interrupt Detection unit (IDU), Interrupt Identification (IIR) registers and Interrupt request (IRQ) generation unit as shown in Figure 4.

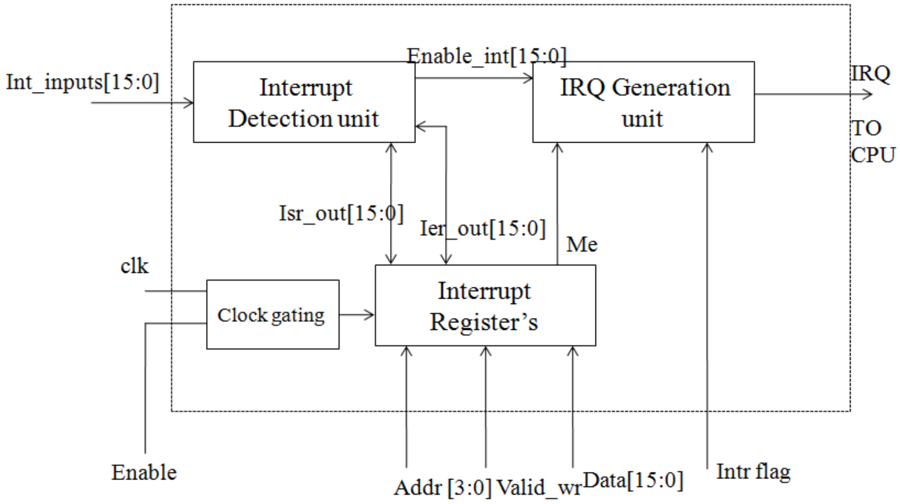


Figure 4. Block diagram shows Interrupt control operation with clock gating technique

Interrupt detection unit recognizes the interrupts upon the negative clock edge of the clock signal. Once interrupts are identified then check for the corresponding interrupt input masked or not. An unmasked interrupt input set, the corresponding bit in the interrupt status register (ISR) are available. IRQ generation unit generates the interrupt request by using the interrupt vector register contents. Interrupt request reaches the RISCPU send an acknowledgement signal.

3.2. Transmitter

The RISA has independent transmitter and receiver modules. The transmitter [12], [13], [14] portion of this Processor accepts a byte of data from the parallel data and transmits it as serial data on the TXD port. It consists of different blocks such as address decoder, transmit buffer, transmit fifo, transmit controller and a parallel to serial converter. Applying clock gating to the transmitter section can view in Figure 5.

The operation of the transmitter is as follows. Initially, if want to send a data first active all the blocks using clock gating. Once gating signal active and then send a 32-bit packet data in serial form and enable the transmitter.

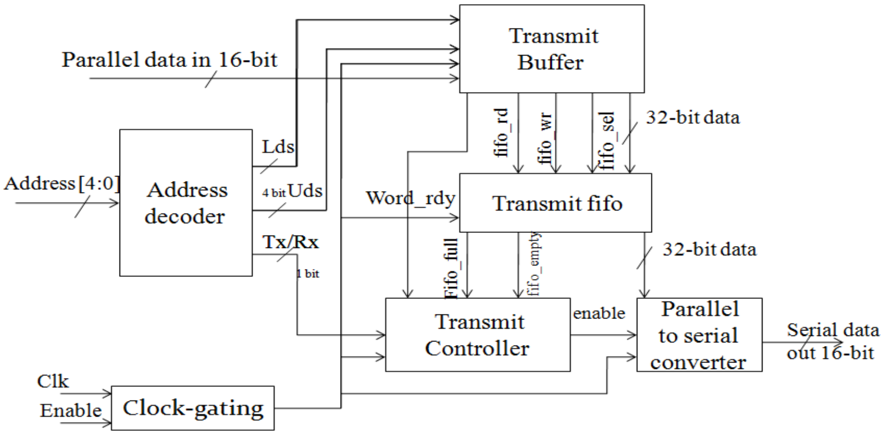


Figure 5. Block diagram of transmitter module with clock gating technique

This can be done by writing or loading the proper control word address into the SFR. After writing (loading) the address into the SFR then the transmitter will be reads the address and decoding using control signals such as lower data select (LDS) and upper data select (UDS). If the fifo_wr signal is asserted and data is available on the data bus (16-bit) based on the signals either LDS and UDS is loaded into to buffer. If fifo_rd is asserted then the contents of the status register are send output data. When the buffer is filled by word_rdy signal loads the 32-bit data into the transmit fifo. The transmit controller generates necessary control signals (such as tx/rx, word_rdy, fifo_full, fifo_empty). Then the enable signal is asserted the 32-bit data in the transmit fifo is loaded into the parallel to serial converter and then data will be transmitted into desried locations.

3.3. Receiver

The receiver module is completely reverse operation of the transmitter. The RISA contains independent transmitter and receiver modules .The receiver [12], [13], [14] portion of this Processor accepts byte of data from the serial data and convert it as parallel data on the RXD port. It consists of different blocks such as address decoder, serial to parallel converter, receive fifo, receive controller and receiver buffer. Receiver module of RISA as shown in Figure 6.

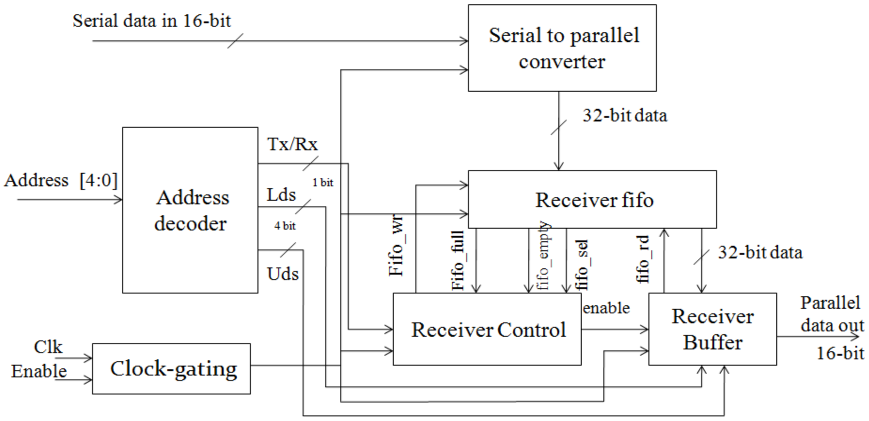


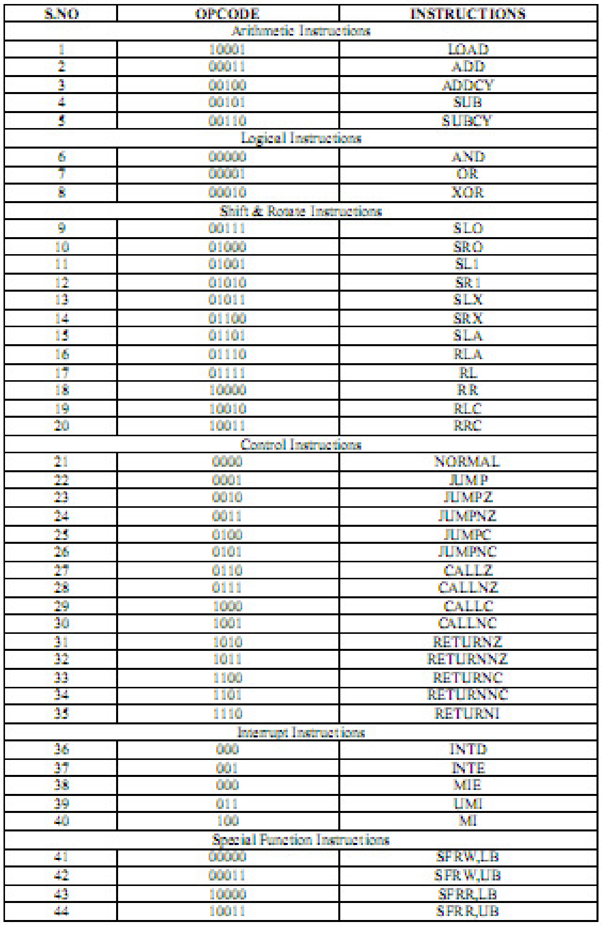
Figure 6. Block diagram of receiver module with clock gating technique

The operation of the receiver module is as follows. Initially, if want to receive a serial data from sensor locations first active all the blocks using clock gating. Once receiver is enabled, 32-bit serial data is converted into 32-bit parallel data by using serial to parallel converter and then data will be stored in receiver fifo. At this time the receiver buffer will generate fifo_wr signal. The receiver controller generates necessary control signals connected by blocks such as tx/rx, fifo_sel, fifo_full and fifo_empty. The 32-bit parallel data can be divided (lower and upper) into two 16-bit words after fifo_ rd signal is high.

4. INSTRUCTION SET

Reduced instruction set architecture [11], [18], include arithmetic instructions, logical instructions, program control instructions, shift and rotate instructions, input/output instructions, interrupt instructions and special function instructions are shown in table 1.

Table 1. Instruction Set of RISA



S.NO	OPCODE	INSTRUCTIONS
Arithmetic Instructions		
1	10001	LOAD
2	00011	ADD
3	00100	ADDCY
4	00101	SUB
5	00110	SUBCY
Logical Instructions		
6	00000	AND
7	00001	OR
8	00010	XOR
Shift & Rotate Instructions		
9	00111	SLO
10	01000	SRO
11	01001	SL1
12	01010	SR1
13	01011	SLX
14	01100	SRX
15	01101	SLA
16	01110	RLA
17	01111	RL
18	10000	RR
19	10010	RLC
20	10011	RRC
Control Instructions		
21	0000	NORMAL
22	0001	JUMP
23	0010	JUMPZ
24	0011	JUMPNZ
25	0100	JUMPC
26	0101	JUMPNC
27	0110	CALLZ
28	0111	CALLNZ
29	1000	CALLC
30	1001	CALLNC
31	1010	RETURNZ
32	1011	RETURNNZ
33	1100	RETURNC
34	1101	RETURNNC
35	1110	RETURNI
Interrupt Instructions		
36	000	INTD
37	001	INTE
38	000	MIE
39	011	UMI
40	100	MI
Special Function Instructions		
41	00000	SFRW,LB
42	00011	SFRW,UB
43	10000	SFRR,LB
44	10011	SFRR,UB

The arithmetic instructions are used to handle the data coming from various devices. The logical instructions are used for testing of sensor data. The shift and rotate instructions are used to flow of execution data from processor to other external devices. The control instructions are used to flow of execution in a processor. The interrupt instructions are used to masking and unmasking of interrupts coming from various peripherals. Some specific instructions are used to transmit and receive data from sensor locations.

5. IMPLEMENTATION

The RISA was implemented using Xilinx platform on Virtex4 FPGA family in VerilogHDL. Some application specific instructions and interrupt instructions are designed along with general purpose instruction set. These instructions are used to reduce the heavy load of the RISA architecture during the transmission and reception of data from sensor locations.

The flow chart for the clock gating is shown in Figure 7. When the clock signal and clock gated control signal asserted, it allows clock signal to the required module otherwise block the particular module in architecture.

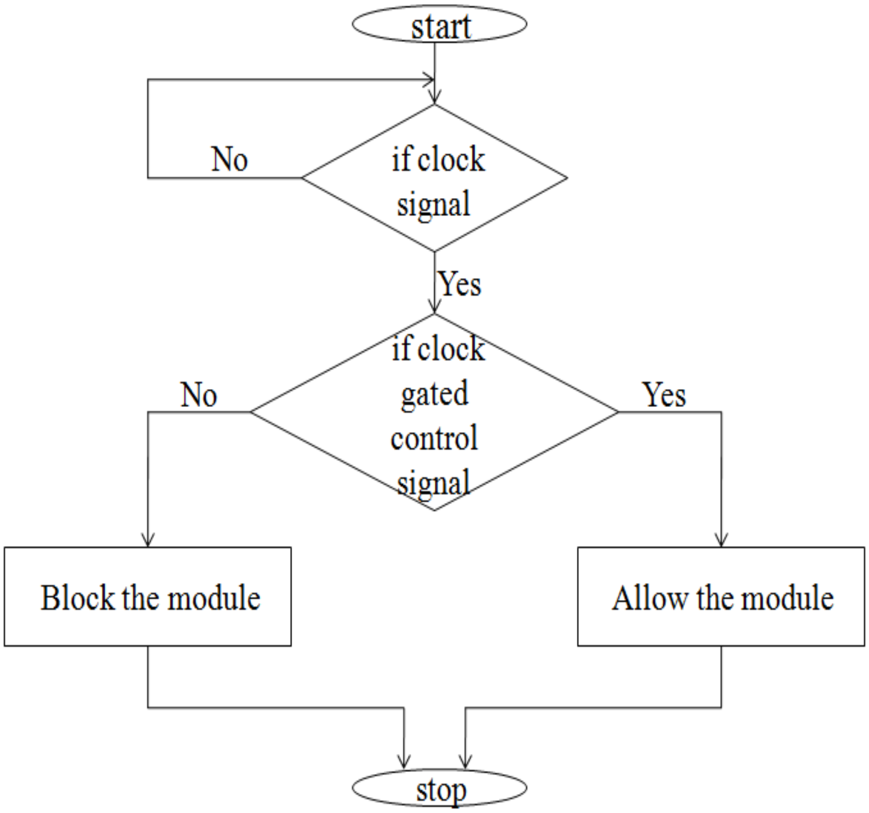


Figure 7. Flowchart for clock gating technique

The flow chart for the interrupt controller with clock gating is shown in the Figure 8. If clock gated signal is active then select interrupt controller [16] module. Various peripherals (internal or external devices) want the services provided by RISCPU. They generate an interrupts to interrupt controller. During negative cycle of this clock signal, the interrupt detection unit identify the interrupts coming from various devices and also verify the interrupt is masked or not. This information is present in the Interrupt enable register (IER). It holds the interrupt enable signal (enable_int). This information is stored in the Interrupt status register (ISR). Interrupt request are generated when master enable bit is set and then check the status of interrupt flag. Once interrupt flag bit is high, interrupt request is generated. RISCPU waiting for the acknowledgement.

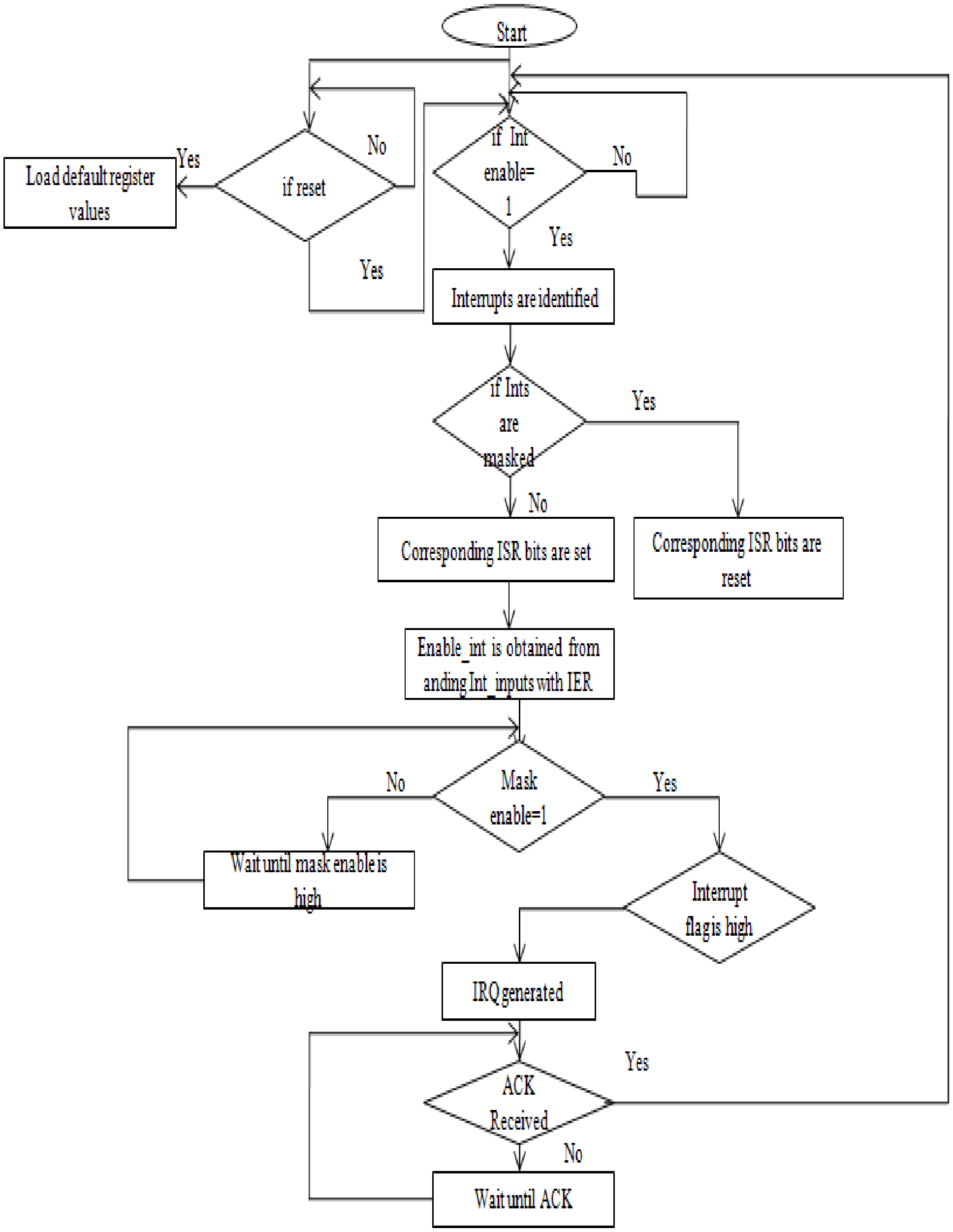

Start
if reset
Yes
No
Yes
Load default register values
if Int enable= 1
No
Yes
Interrupts are identified
if Ints are masked
Yes
No
Corresponding ISR bits are set
Corresponding ISR bits are reset
Enable_int is obtained from anding Int_inputs with IER
Mask enable=1
No
Yes
Wait until mask enable is high
Interrupt flag is high
IRQ generated
ACK Received
Yes
No
Wait until ACK

Figure 8. Flow chart for Interrupt controller with clock gating

The flow chart for the transmitter module with clock gating is shown in Figure 9.

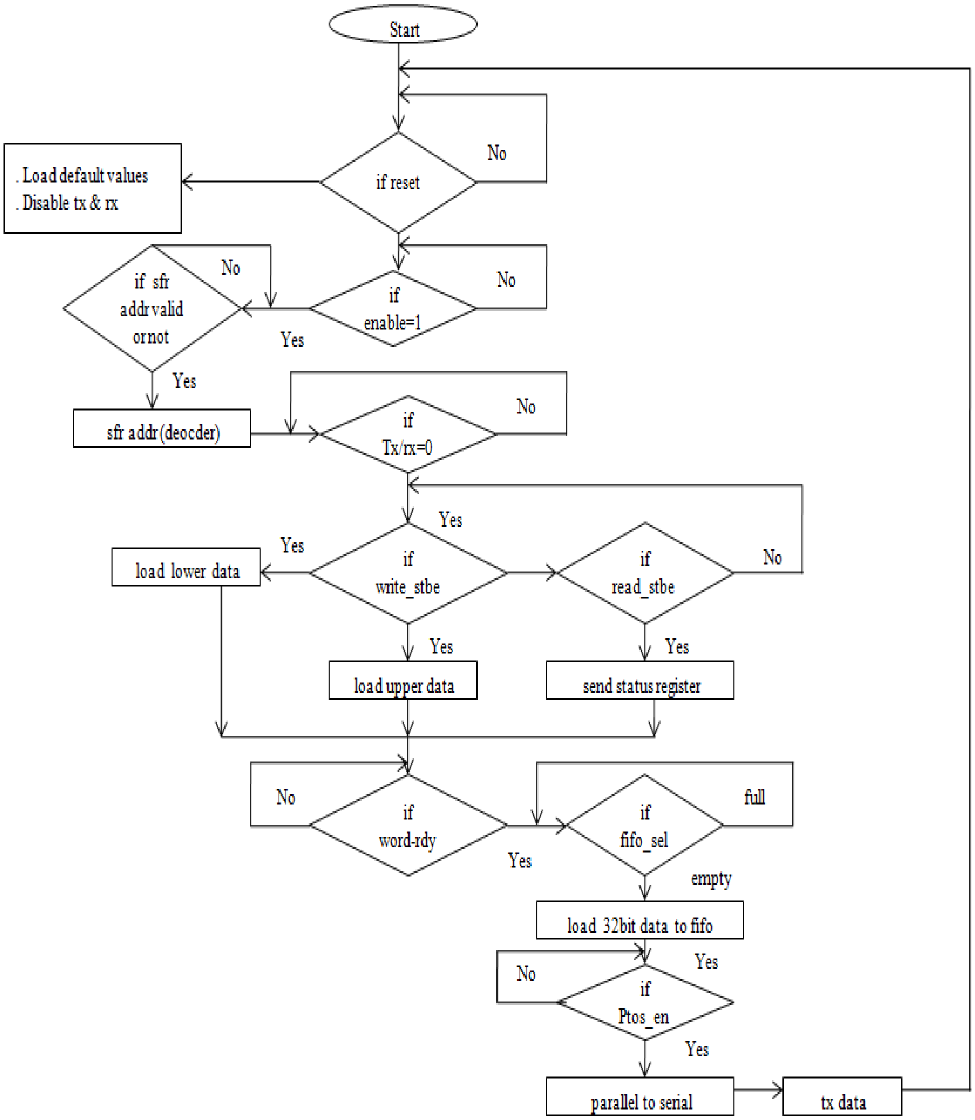


Figure 9. Flow chart for Transmitter with clock gating

At the beginning, tx_rx signal is enabled and the values are loaded into the registers after reset signal is high. If the reset signal is low, then the transmitter module waits for the valid special function register address. If the valid special function address is available then the addresses will be decoded and the tx_rx signal is disabled. If the write_ strobe signal is high, upper or lower byte is loaded into the transmitter buffer. When read_strobe signal is high, the contents of status register (16-bit) are sent to the output port. If word_rdy signal is high, the 32-bit data in the transmitter fifo is loaded into the parallel to serial converter and the data will be transmitted to sensor locations when ptos_en signal is high

The flow chart for the receiver module with clock gating is shown in figure 10.

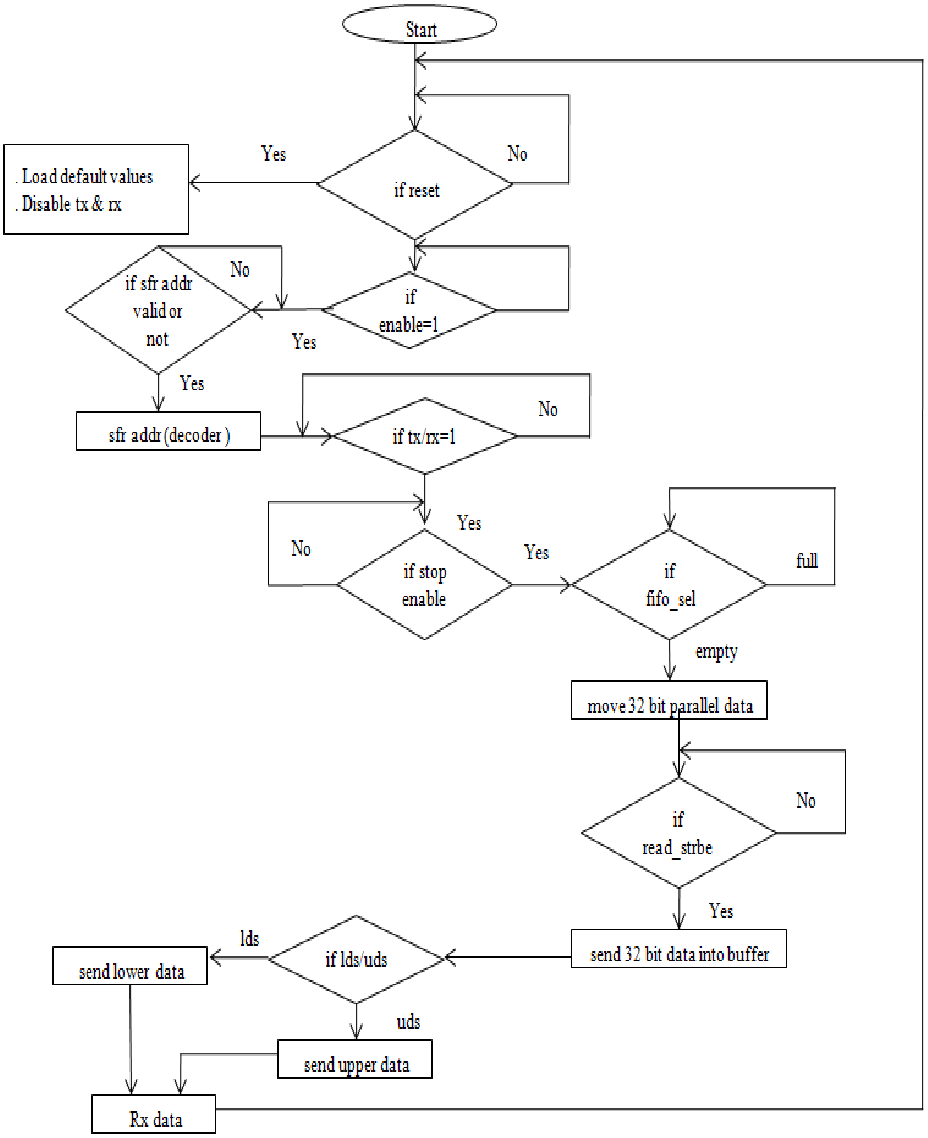


Figure 10. Flow chart for Receiver with clock gating

At the beginning, tx_rx signal is disabled and the values are loaded into the registers after reset signal is high. If the reset signal is low, then the receiver module waits for the valid special function register (SFR) address. If the valid SFR address is available then addresses will be decoded and the tx_rx signal is enabled. When the stop_enable signal is high, the receiver module starts receiving the serial data from sensor locations. Then this received serial data is loaded into the receiver fifo based on the fifo control signals. If the fifo_sel signal is full then the serial receiver waits till at least one position in the fifo_full and it is fifo_empty signals, then the 32- bit parallel data is immediately loaded into the receiver fifo in a receiver module. When the read_strobe signal is high, this 32-bit data is loaded into the 32-bit receiver buffer register. Then based on the decoded signals such as LDS or UDS, the corresponding lower or upper 16-bit data is sent to the output port.

6. RESULTS

The top level module with input and output signals of the RISA with clock-gating is shown in Figure 11 and the RTL schematics of the Interrupt controller, transmitter and receiver are shown in the Figures 12,13 and 14 respectively.

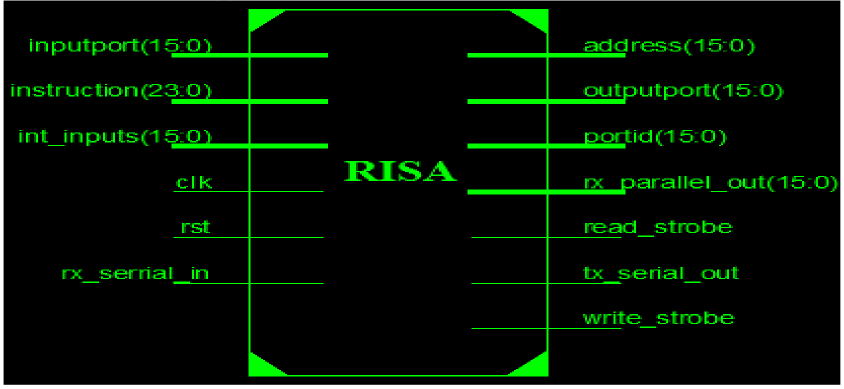


Figure 11. Signals of RISA

The schematic diagram of the Interrupt Controller of RISA is shown in Figure 12. The Interrupt controller consists of a Register block, interrupt detection unit and Interrupt request generation unit. It shows input & output signals (pins) of the Interrupt Controller with clock gating and connections between the blocks.

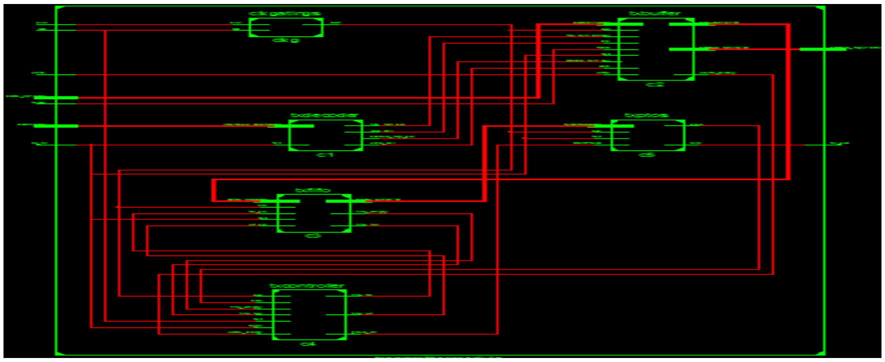


Figure 12. Schematic diagram of the interrupt controller of RISA

The schematic diagram of the transmitter module of RISA is shown in Figure 13. It consists of address decoder, transmit buffer, transmit fifo, transmit controller, and parallel to serial converter. It shows input & output signals (pins) of the transmitter module with clock gating and connections between the blocks.

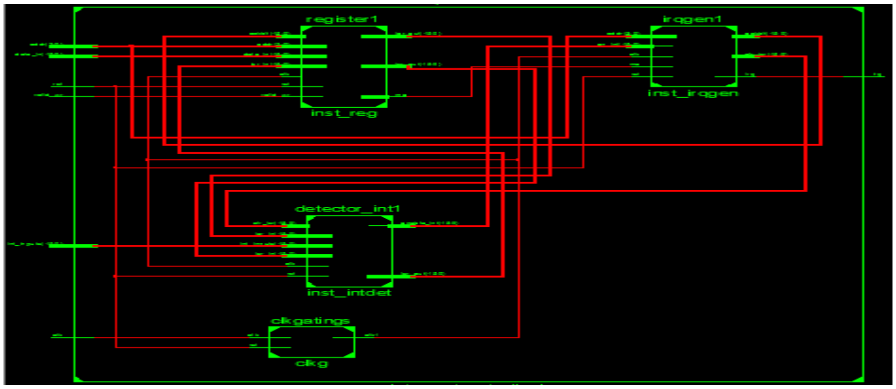


Figure 13. Schematic diagram of the transmitter module of RISA

The schematic diagram of the receiver module of RISA is shown in Figure 14. The receiver consists of address decoder, Receive buffer, Receive fifo, receive controller, and serial to parallel converter. It shows connection between the different blocks and the input & output signals of the receiver module with clock gating.

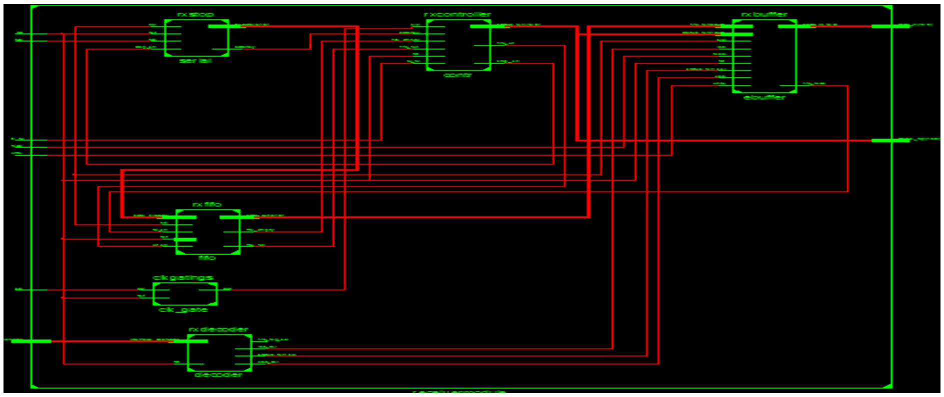


Figure 14. Schematic diagram of receiver module of RISA

Figure 15 shows the simulation result of Interrupt controller with clock gating. If reset signal asserted high then loaded default values into the registers in interrupt controller module. Initially giving int_inputs data is 0051h loaded. By using 000000h instruction, disables the interrupt request. Even though int_inputs are active; RISA does not receives any interrupt signal because of master enable (ME) bit is low. If ME output is high, interrupts are received from RISCPU.

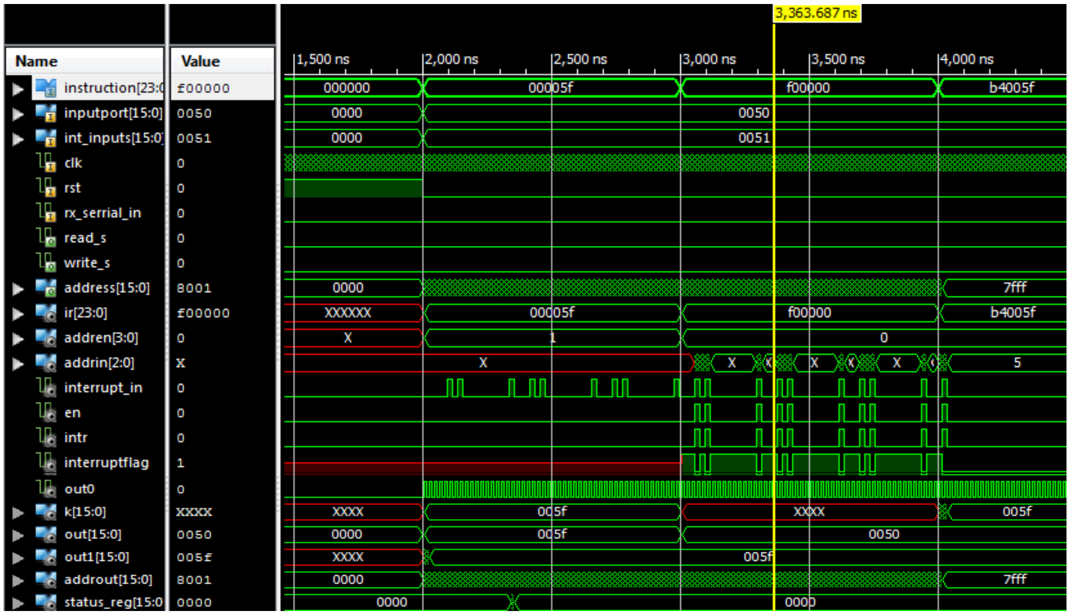


Figure 15. Simulation results for interrupt controller and RISCPU

The simulation result of Interrupt controller, transmitter shown in Figure 16. Initially generate interrupts coming from various devices using Interrupt controller using Instruction set. Once generates interrupt, the data (0005h) is loaded into the register. When the tx/rx signal is asserted low the transmitter starts transmit data. OUT instruction are used to sends the data into the output port. In this architecture sends the data in packet format using two 16-bit words. The lower 16-bit data indicates the actual sensor data while upper 16-bit data indicates the payload attached to the data. To send the lower byte and upper byte data using two specific instructions are used. When ptos_en (parallel to serial converter) signal is high, the data will be transmitted into desired locations.

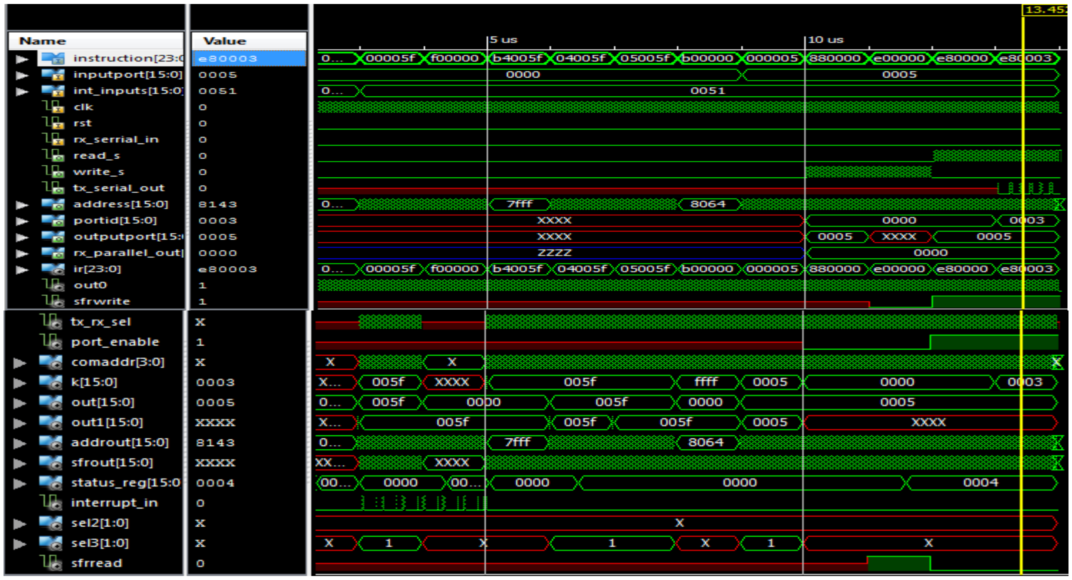


Figure 16. Simulation results for Interrupt controller, transmitter and RISCPU

The figure 17 shows results of simulation of interrupt controller, receiver and RISCPU. When the tx/rx signal is asserted high the receiver starts receiving the serial data. The receiver fifo is filled with 32 bit data received from sensor locations. At this time if read _strobe signal is asserted high, then the data is loaded onto the receiver buffer. The upper data or lower data (16 bit) word signals are selected and then data will be obtained from output port.

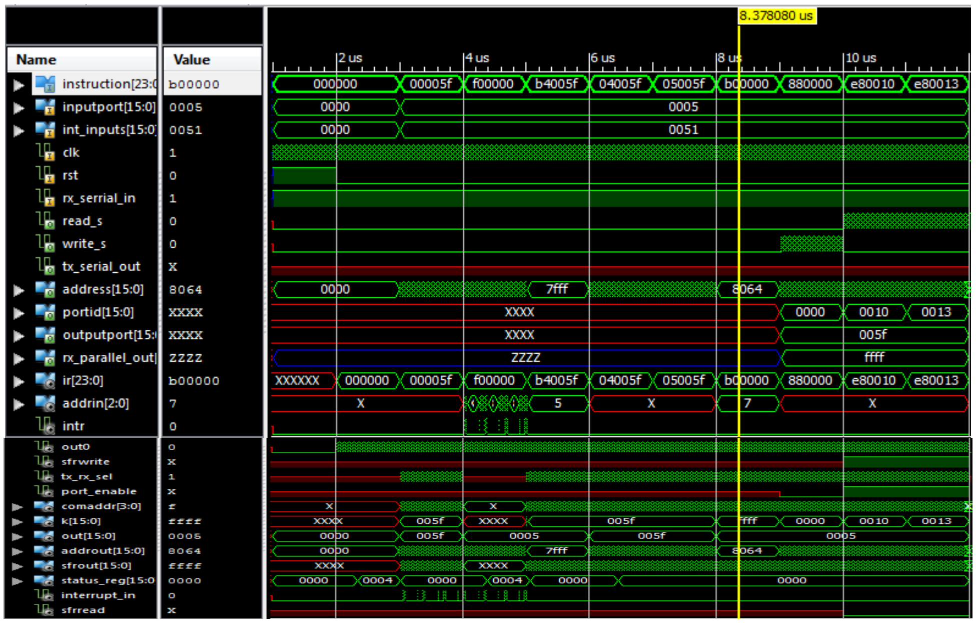


Figure 17. Simulation results for Interrupt controller, Receiver and RISCPU

The comparison of different modules used in a RISA processor is compared with ASIP [9] core at 1.2v and operating frequency of 1GHz is shown in Figure 18.

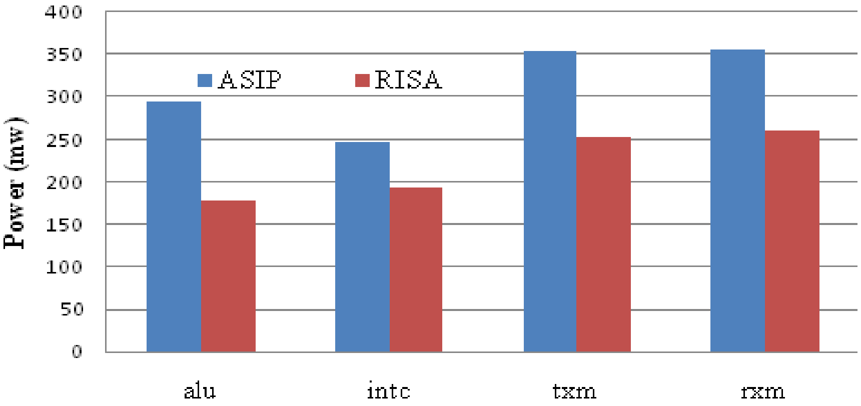


Figure 18. Power consumption of various modules in RISA and ASIP [9]

The comparison of the RISA and ASIP [9] cores in the form of power consumption using Xpower analyzer tool in a Xilinx. In the Figure 19 shows the variation of the curves indicates that the operating voltage is increasing then the power consumption of the ASIP [9] core increases more rapidly when compared with RISA power at 1GHz frequency.

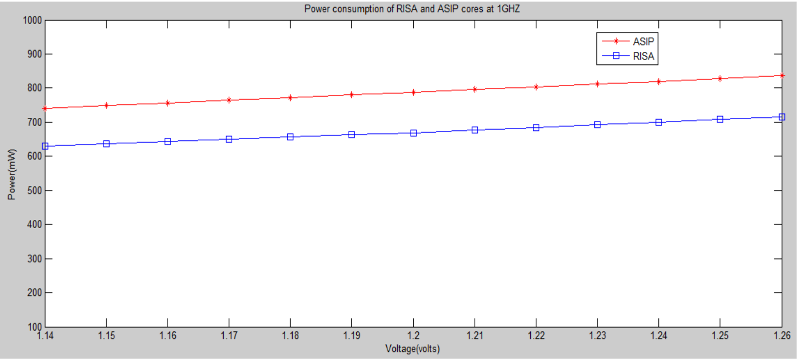


Figure 19. Power consumption of RISA and ASIP [9] cores at 1GHz

In the Figure 20, the deviation of the curves indicates that as the operating frequency is increasing the power consumption of the ASIP core increases more rapidly when compared with RISA power at 1.2volts.

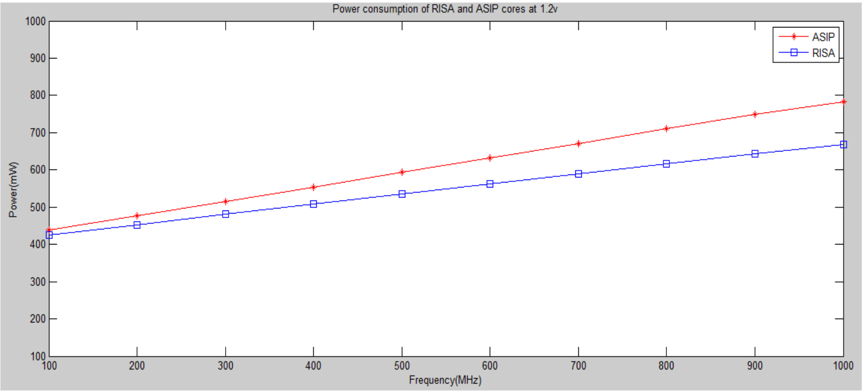
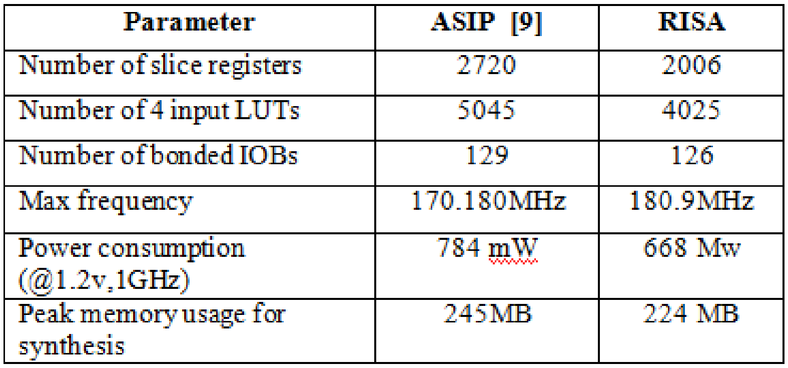


Figure 20. Power consumption of RISA and ASIP [9] cores at 1.2v

A comparison of different device utilization parameters of ASIP and RISA cores is shown in TABLE 7. The table also gives the specifications of the RISA CORE.

Table 7. Comparison of device utilization parameters for ASIP and RISA



Parameter	ASIP [9]	RISA
Number of slice registers	2720	2006
Number of 4 input LUTs	5045	4025
Number of bonded IOBs	129	126
Max frequency	170.180MHz	180.9MHz
Power consumption (@1.2v,1GHz)	784 mW	668 Mw
Peak memory usage for synthesis	245MB	224 MB

6. CONCLUSION

In this paper clock gating technique is applied to optimize the power of Reduced Instruction Set Architecture (RISA) employing RISC architecture in a single chip. In this architecture, number of interrupts are increased and provided serial communication on this chip itself. The designed Architecture power saving upto 19.83%; when processor operated at 1.2v and operating frequency of 1GHz. The advantage of this architecture, it can handle an interrupt faster, serial data at high speed and effectively. It occupies less area and less power consumption.

REFERENCES

[1] Jagrit Kathuria,M.Ayoub khan, Arti noor, "A review of clock gating techniques", International Journal of Electronics and Communication Engineering, , Vol. 1, No. 2 pp.106-114,, Aug 2011.

[2] Khaja Mujeebuddin Quadry, Dr. Syed Abdul Sattar, ''Design of 16 bit low power processor", (IJCSIS), International Journal of Computer Science and Information Security Vol. 10, No. 6, pp.67-71 June 2012, ISSN 1947-5500

[3] M.Kamaraju, K.Lal Kishore, A.V.N.Tilak, "Power optimized programmable embedded Controller", International Journal of Computer Networks & Communications (IJCNC), Vol.2, No.4,pp 97-107 July 2010

[4] P.Sreenivasulu, Krishnna veni ,Dr. K.Srinivasa Rao and Dr.A.VinayaBabu, "Low Power Design Techniques Of Cmos Digital Circuits" International journal of Electronics and Communication Engineering &Technology (IJECET), , pp. 199 -208, Published by IAEME Volume3, Issue2, 2012

[5] J. Shinde and S. S. Salankar, "Clock gating-A power optimizin technique for VLSI circuits," in Proc. Annual IEEE India Conference (INDICON), pp. 1-4, 2011.

[6] Xiaotao Chang, Mingming Zhang, Ge Zhang, Zhimni Zhang and Jim Wang,"Adaptive Clock Gating Technique for Low Power IP Core in SoC Design",IEEE International Symposiumon Circuits And Systems,May(2007),pp. 2120 – 2123, 2007.

[7] IBM, Multiprocessor Interrupt controller Data Book, March, 2006.

[8] Xilinx OPB Interrupt Controller (v1.00c), January 2005.

[9] M.Kamaraju,K.LalKishore,A.V.N.Tilak, "A Novel Implementation of Application Specific Instruction-set processor(ASIP) using Verilog" International Journal of Science, Engineering and Technology(pISSN 2010-376X, eISSN 2010-3778), Issue 59, Nov 2011, pp 382-387

[10] Jin Soo Kim Sunwoo, Myung H. Sunwoo, "Three low power ASIP processor designs for communications, video, and audio applications", International Conference on Design & Technology of Integrated Systems in Nanoscale Era, 2007. DTIS, pp. 241-244.

[11] Van prate.J, Goossens.G, Lanneer.D, and De man.H "Instruction set definition and instruction set selection for ASIPs", In Proc. of the 7th International Symposium on High-Level Synthesis 1994, pp. 11–16.

[12] Mohan .R, Land.I, "Building Intergrated ARINC 429 Interfaces using an FPGA", Actel Corporation, Mountain View, California, 2005

[13] ARINC Specifications 429P1-15, Sep. 1, 1995.

[14] ARINC Specifications 429p2-15, March 6, 1996.

[15] Wei Chipni, Li Ahao lin, Zheng Qingwei,Ye Jianfei, and Li Shenglong, "Design of a configurable multichannel interrupt controller",Second Pacific- Asia Conference on Circuits, Communications and System, vol. 1,Aug 2010, pp. 327–330.

[16] Tumeo, M. Branca, L. Camerini, M. Monchiero,G. Palermo, F. Ferrandi and D. Sciuto, "An Interrupt Controller for FPGA-based Multiprocessors", International Conference on Embedded Computer Systems: Architures, Modeling and Simulation, July 2007,pp. 82-87

[17] Khaja Mujeebuddin Quadry, Dr. Syed Abdul Sattar, Dr. K. Soundara Rajan, "Design Of 16 Bit Low Power Processor Using Clock Gating Technique'', International Journal Of Electronics And Communication Engineering & Technology (IJECET), Volume 3, Issue 3, October- December (2012), pp. 333-340

[18] Simple Scalar Portable Instruction Set Architecture (PISA).[http://www.simplescalar.com/].

[19] M.Kamaraju, Praveen V N Desu,"FPGA Implementation of RISA IC using Verilog," Journal of Innovation in Electronics & Communication[ISSN: 2249-9946], Vol.2, Issue 2, Jan 2012, pp.189-193

[20] Motorola Inc.”M6800 8/16/32 – bit Microprocessor”, available at http://www.freescale.com/files/32bit/doc/ref_manual/MC68000UM.pdf.

[21] ASIP Meister. http://www.eda-meister.org/asipmeister.

Authors Biography

M.Kamaraju[1] obtained his Bachelor’s Degree & Master’s degree from Andhra University and Ph.D from JNTUH, Hyderabad in the area of Low Power VLSI Design; Areas of interest are Microprocessors, Microcontrollers, Digital system Design, Embedded System Design, Low Power VLSI Design. He published 59 technical papers in national/ international journals/conferences. He reviewed number of papers for international journal and conferences. He is a Fellow of IETE and IE and member of IEEE. Presently working as Professor & Head of ECE Department, Gudlavalleru Engineering College, Gudlavalleru, India and chairman of IETE, Vijayawada centre.

G.Chinavenkateswararao[2] obtained his Bachelor’s Degree from Sasi Institute of Technology & Engineering, Tadepalligudem. His Areas of interest in Microprocessors, Microcontrollers, Embedded System Design, Low Power VLSI Design, Digital Signal Processors and Digital System Design. He published one review paper on Low Power Instruction Set Architecture for Aircraft Applications Using Clock Gating Technique. He is a member of IETE. Presently he is doing M.Tech Digital Electronics and Communcation Systems at Dept.of ECE, Gudlavalleru Engineering College, Gudlavalleru, India.