POWER OPTIMIZATION OF APPLICATION SPECIFIC PROCESSOR USING CLOCK GATING TECHNIQUE

Shaik Chinna Baji[1], Kadiyam Tirumala Rao[2], M.Muzammil Parvez[3]

***ABSTRACT*--**Low power processors are realization of portable electronic devices, in which power consumption is an important consideration of today. Low power consumption reduces heat dissipation, improves battery life and also steadfastness. The proposed architecture has a dynamic control unit which consists of 16 bit Interrupt controller, Transmitter, Receiver, program flow controller and I/O controller**.** This architecture can handle multiple interrupts and performs serial communication effectively. It can support RISC (Reduced Instruction Set Computer) concepts. Clock Gating is a well known technique to reduce the power consumption in this proposed Application Specific Processor (ASP). This entire architecture captured using Verilog HDL and implemented on FPGA using Xilinx tools.

Keywords----- power consumption, RISC, FPGA, Interrupts, Clock-gating, Xilinx Xpower tools.

I. 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. Embedded systems today are operated with battery; the core of the embedded systems may be the processor or controller. If it is consuming more power, then life of battery also decreases. Particularly in low power design, the core of the system consumes low power. It reduces the overall power dissipation.

An Application Specific Processor (ASP) is a processor that has been embedded into a device. It can be programmed to interact with different pieces of hardware. Performance wise, an embedded processor [3] can have better performance than a microcontroller, but does not have as much performance as a general-purpose microprocessor.

This paper describes the architecture of ASP with 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. The technique mainly concentrates on reducing power and also establishing serial communication between two systems.

In the earlier, designers treat the clock signal should not be disabled or disturbed but clock signal is a major source for power dissipation and it is a dynamic in nature because clock signal is fed into several blocks in the processor. Because all the modules usage varies within and across a processor, the modules which are unused all the time and give a chance to reducing the power consumption of unused module.

Clock-gating [1], [4], [5], [6] is a technique where the clock signal is prevented from reaching the various modules of the processor. The nonappearance of the clock signal prevents any register and or flip-flop from varying their value. Hence the input to any combinational logic circuit remains unchanged, and no switching action takes place in those circuits. This architecture consists of the ALU, Port Controller, Interrupt Controller, Transmitter and the Receiver to apply clock gating technique. These are the main modules in our architecture in terms of number of logic gates. The Control Unit is used for generating the clock gating signal based on the current instruction.

The general purpose processors provide one or more interrupt pins that allow external devices to request the services provided by CPU. Consider a case in which processor can handle a large number of interrupts which are come from external devices. So this paper describes about separate interrupt controller which is interfaced to the processor and also separate transmitter and receiver modules are interfaced. This interfacing increases the complexity of design and speed of the processor.

The following sections will provide a brief overview of architecture of the ASP with clock gating with explanation about the implementation. Then brief description about the Control unit, Interrupt controller, Transmitter and Receiver sections of the ASP with clock gating are given. The first section describes about the overall architecture of ASP. Later the individual modules in the system explained. Finally few notes on simulation results.

II. APPLICATION SPECIFIC PROCESSOR ARCHITECTURE

The proposed ASP Architecture mainly consists of RISCPU, Interrupt controller, Port controller, Program flow controller, Transmitter and Receiver and its architecture is shown in the Fig. 1. These blocks are connected by 16-bit internal buses.

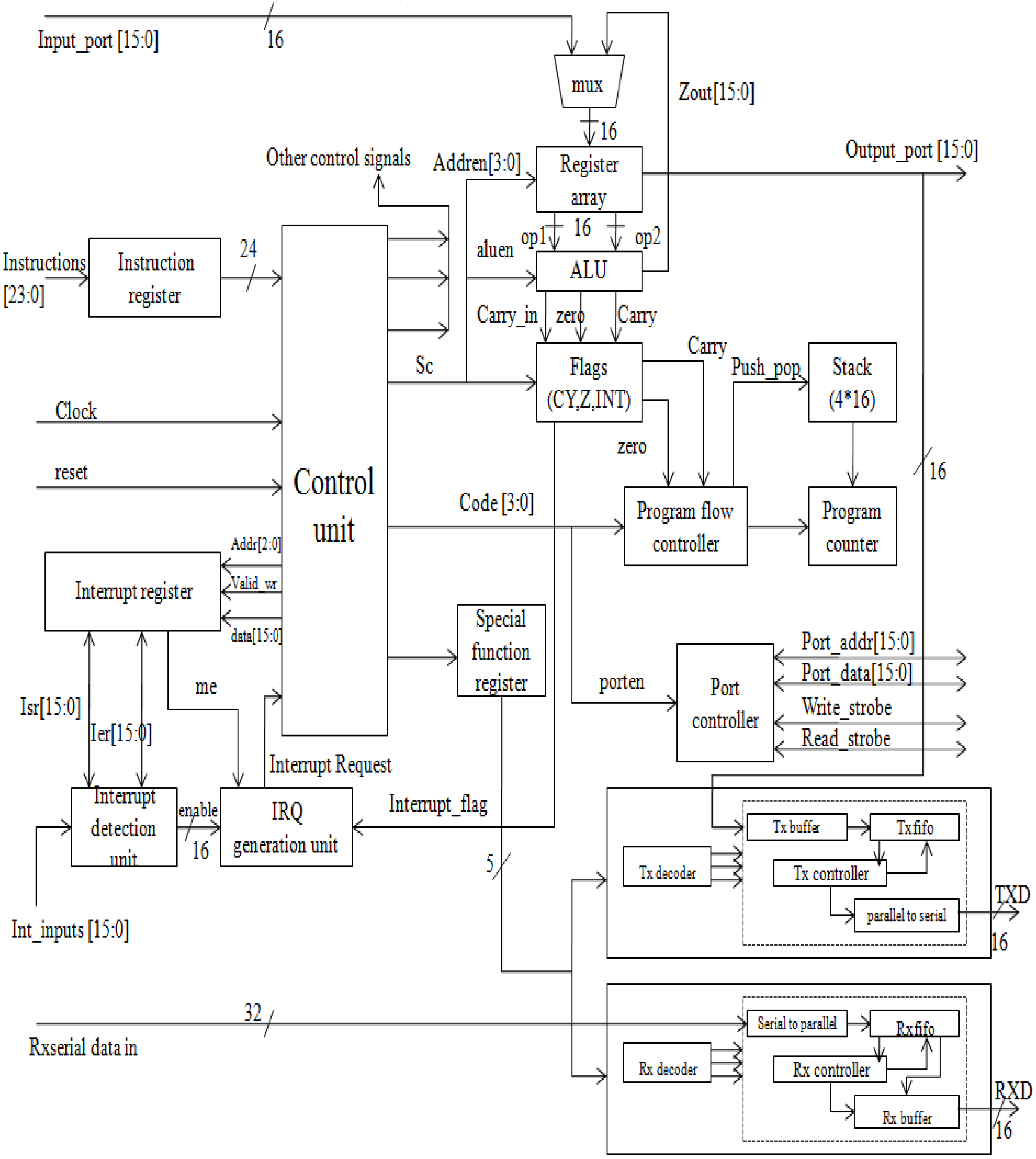

Fig. 1. ARCHITECTURE of ASP CORE

The Reduced instruction set central processing unit (RISCPU) of ASP consists of 16-bit address and data buses. The ALU (Arithmetic and Logical Unit) is a fixed integer type which perform 16-bit arithmetic and logical operations on the sensor data and also useful for determine the flow of execution when branch and jump instructions. The RISCPU also consists of 8 general purpose registers each of 16 bits.

III. CONTROL UNIT

The control unit provides all of control signals to regulate the data traffic and necessary signals to perform the desired functions. The Reduced instruction set central processing unit (RISCPU) of ASP consists of 16-bit address and data buses. The ALU (Arithmetic and Logical Unit) is a fixed integer type which perform 16-bit arithmetic and logical operations on the sensor data and also useful for determine the flow of execution when branch and jump instructions. The RISCPU also consists of 8 general purpose registers each of 16 bits.

The instruction length of the ASP is 24-bit wide. The RISCPU has three flags namely carry, zero and interrupt flags. Both carry and zero flags are affected only during the execution of arithmetic and logical instructions. CPU checks the interrupt flag after completion of every instruction to know whether interrupt is available or not. Along with this a stack is used to store up to four (16-bit) addresses during interrupt and branch related instructions.

A. INTERRUPT CONTROLLER

Interrupt controller has multi read port and single write port. Read and write operations are performing during the positive edge and negative edge of the clock. Port controller takes care of the read and writes operation. An16-bit address value provided on the “port” bus together with a ‘read and write strobe’ signal indicated the accessed port. The port address are used to supply in the program as an absolute value or specified indirectly as the contents of any of the eight registers. There are some specific instructions useful for the controlling of interrupt controller present in the ASP.

It contains Interrupt Detection unit (IDU), Interrupt Identification (IIR) registers and Interrupt request (IRQ) generation unit as shown in fig. 1. Interrupts are identified by interrupt detection unit during the negative clock edge of the clock. Whenever interrupts are detected it checks for the corresponding interrupt input masked or not.Unmasked interrupt input set the corresponding bit in the interrupt status register. IRQ generation unit generates the interrupt request by using the interrupt vector register contents. Interrupt request reaches the RISCPU sends an acknowledgement signal.

B. TRANSMITTER

The ASP 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. This consists of different blocks such as transmit buffer , transmit fifo, address decoder, transmit controller and a parallel to serial converter as shown in fig.1.

The function of the transmitter is as follows. Initially, if we want to send a data first we have to active all the blocks using clock gating. Once gating signal active and then it sends a 32-bit packet data in serial form and enable the transmitter. This can be done by writing or loading the proper address into the SFR. 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 sends output data. When the buffer is filled by word_rdy signal it loads the 32-bit data into the txfifo. The transmit controller generates essential control signals (tx/rx, word_rdy, fifo_full, fifo_empty). Then the enable signal sends the 32-bit data in the txfifo is loaded into the parallel to serial converter and then data will be transmitted.

C. RECEIVER

The receiver performs the reverse operation of the transmitter. The Receiver architecture contains independent transmitter module .The receiver [12], [13], [14] of this Processor accepts byte of data from the serial data and converts as parallel data on the RXD line. It consists of different blocks such as serial to parallel converter, receive fifo, address decoder, receive controller and receiver buffer as shown in fig.1.

The operation of the receiver is as follows. Initially, if it has to send a data first it activates all the blocks using clock gating. To receive the serial data, first the receiver should be enabled. This can be done by writing the proper control word into the SFR. When the receiver is enabled received through the RXD pin. Then this serial data is converted into parallel data by using serial to parallel converter. If serial to parallel conversion is completed for the received 32-bit data a word- rdy signal is generated. At this time the buffer will generate fifo_wr signal. Then the parallel data is temporarily stored in the receiver FIFO. When FIFO read signal is asserted and based on the upper or lower select signals, either upper or lower 16-bit words onto the data bus.

D. PORT CONTROLLER

The Port controllers take care of the read and write operation. An 8-bit address value provided on the port bus together with a read and write strobe signal indicate the accessed port. The port address is either provided in the program as an absolute value or specified indirectly as the contents of any of the eight registers.

E. PROGRAM FLOW CONTROLLER

Instructions are always stored in successive memory locations. When the processed in the CPU, the instructions are fetched from consecutive memory locations and executed. The program control instructions specify conditions for altering the content of the program counter, while data transfer and manipulation of instructions specify conditions for data-processing operations. Some typical program control instructions are Branch, Jump, Skip, Call, Return and compare.

F. SPECIAL FUNCTION REGISTER

Some application specific instructions are particularly used for the data transfer and received from sensor locations.

Several Instructions for various applications have been proposed in this ASP [10]. The instructions are used to transmit and receive data from senor locations

MOV A,#00000H /Transmit lower data/

MOV B, #000011H /Select transmitter upper data/

MOV C,#100000H /Select Receiver lower data/

MOV D,#100011H /Select Receiver upper data/

IV. RESULTS

The entire architecture implemented using Xilinx platform on Virtex4 FPGA family in VerilogHDL. The top level module with input and output signals of the ASP with clock-gating is shown in Fig. 9 and the RTL schematics of the transmitter and receiver, Interrupt controller are shown in the Fig, 3,4 and 5 respectively

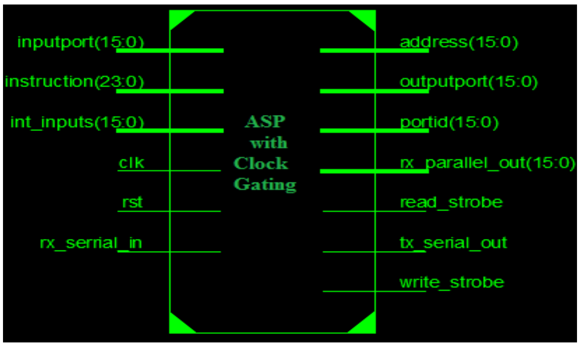

Fig. 2. Signals of ASP with Clock Gating Technique

The RTL schematic of the Interrupt Controller of ASP is shown in Fig. 3. The Interrupt controller consists of a Register block, interrupt detection unit and Interrupt request generation unit. Interconnection between the blocks and the input-output signals of the Interrupt Controller module with clock-gating as shown in the Fig. 3.

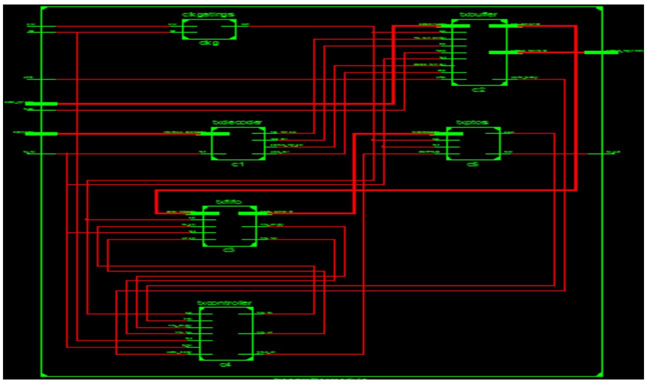

Fig. 3. RTL Schematic of the interrupt controller of ASP

The RTL schematic of the transmitter of ASP is shown in Fig. 4. It consists of address decoder, transmit buffer, transmit fifo, transmit controller, and parallel to serial converter. Interconnection between the blocks and the input-output signals of the transmitter module are also shown in the Fig. 4.

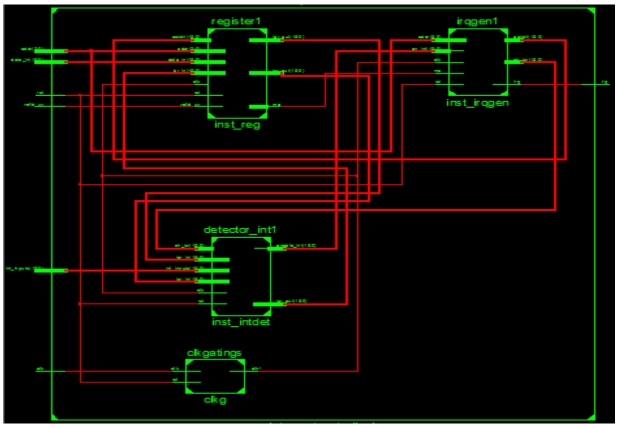

Fig. 4 RTL Schematic of the transmitter module of ASP

The receiver is also similar to that of the transmitter except that the receiver will perform the reverse operation of the transmitter. The receiver consists of address decoder, Receive buffer, Receive fifo, receive controller, and serial to parallel converter. Interconnection between the blocks and the input-output signals of the receiver module are also shown in the Fig. 5.

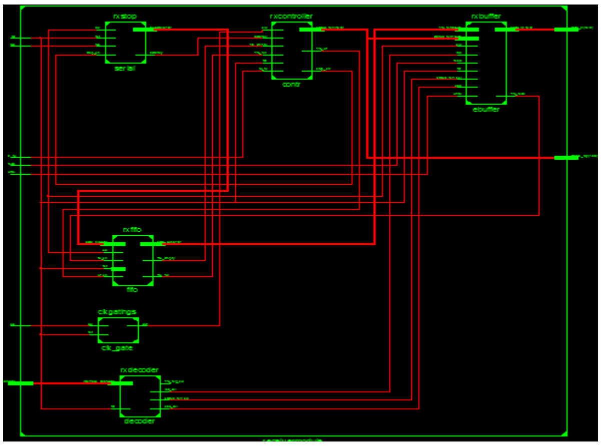

Fig. 5. RTL Schematic of receiver module of ASP

The simulation result of Interrupt controller shown in Fig 6. When the reset is high all the register present in the interrupt controller are loaded with default values. Initially giving int_inputs data is 0051h loaded. By using 000000h instruction it disables the interrupt request. 00005fh instruction are used load default values into the register, b4005fh instruction are used to enable the interrupt module,f000000h instruction are used to active interrupt flag then it is changed from high to low when the interrupt request is received from CPU. RISCPU does not receive any interrupt even though int_inputs are active because ME output is low . If ME output is high, interrupts are received from CPU.

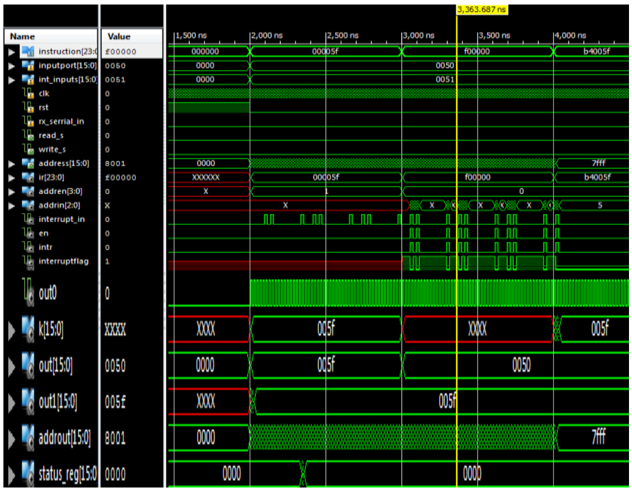

Fig. 6. Simulation results for interrupt controller and RISCPU

The simulation result of Interrupt controller, transmitter shown in Fig 7. Initially generate interrupts coming from various devices using Interrupt controller using Instruction set. Once interrupt is generated, the data (0005h) is loaded into the register. When the tx/rx signal is asserted low the transmitter starts transmit data. To verify that the data was properly loaded in the register the OUT instruction was written. The instruction sends the data to the output port (0000h indicates port address). In the simulation results we can clearly observe the 0005h at the output port. Next in order transmit the data it should be 32-bit format. So the processor has to send two 16-bit words to form a packet. The lower byte indicates the actual sensor data while upper byte indicates the payload attached to the data. In the above program both upper and lower bytes are 0005h. To send the upper byte and lower byte two instructions are written. The data transmission started when ptos_en signal is asserted.

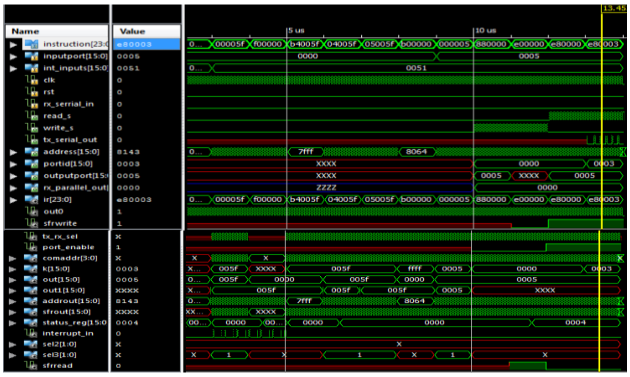

Fig. 7. Simulation results for Interrupt controller, transmitter and RISCPU

The simulation results of top level receiver are shown in Fig 8. When the tx/rx signal is asserted high the receiver starts receiving the serial data. If the total 32 bits are received then the packet is moved into the FIFO. At this time if read strobe signal is asserted then the data is loaded onto the buffer. If lower or upper byte selection signals go active high the corresponding data from the buffer is obtained at the output port. In the simulation results shown the upper data or lower data word is selected.

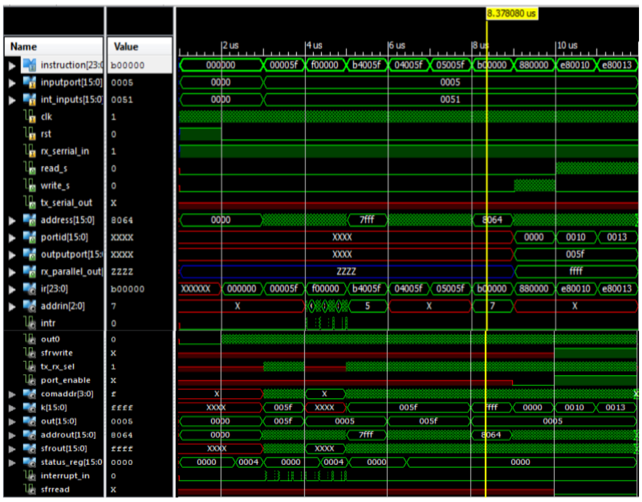

Fig. 9. Simulation results for Interrupt controller, Receiver and RISCPU

Fig 10 represents the place and route diagram of the ASP with clock gating core. The colored area represents routing of all components of the ASP that are placed on the FPGA.

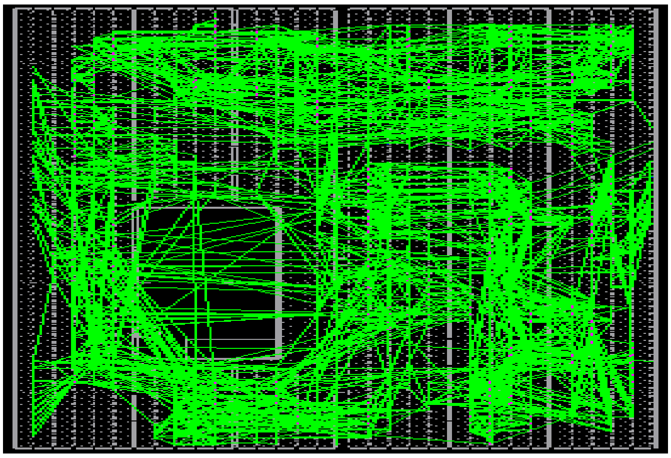

Fig. 10. Place and route diagram of ASP core

The comparison of different modules used in a ASP WITH CLOCK GATING is compared with ASP WITHOUT CLOCK GATING [9] core at 1.2v is shown in Fig 11.

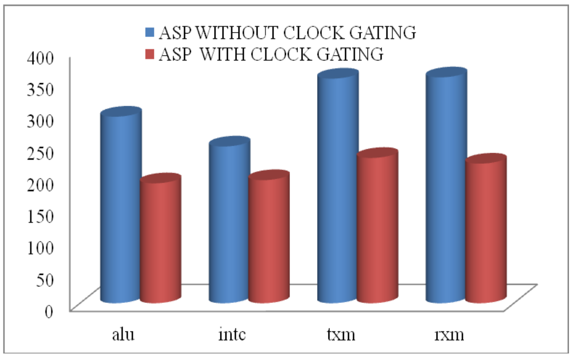

Fig. 11. Comparison various modules of with and without CLOCK GATING of ASP

The Figures 12 and 13 gives comparative view of the power consumption of the with and without clock gating of ASP cores. In the Fig 12, the deviation of the curves indicates that as the operating voltage is increasing the power consumption of the ASIP core increases more rapidly when compared with ASP power. The variation of power with respect to frequency for the ASP and ASIP is shown in Fig. 13.

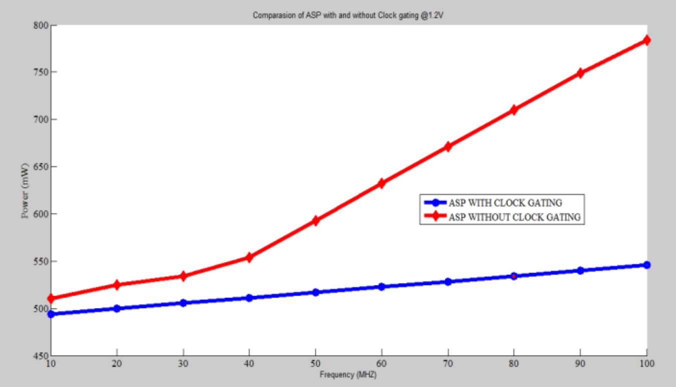

Fig. 12: Comparison of with and without clock gating of ASP @1.2V

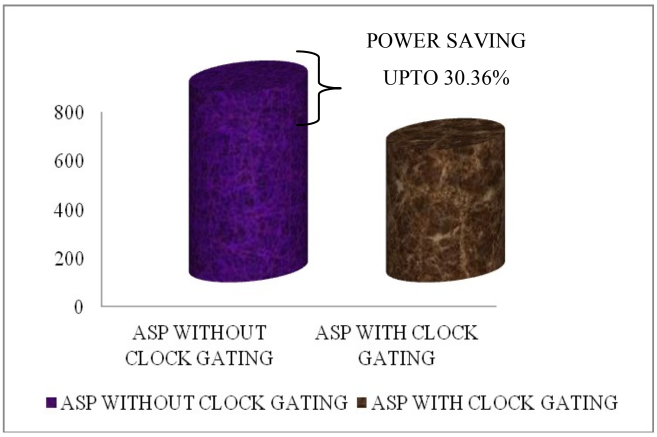

Fig. 13. Power Comparison of with and without Clock Gating of ASP cores at 1.2v

A comparison of different device utilization parameters of with and without clock gating of ASP cores is shown in TABLE 7. The table also gives the specifications of the ASP CORE.

Table7. Comparison of device utilization parameters for with and without Clock Gating of ASP

Parameter	ASIP [9]	ASP with CLOCKGATING
Number of slice registers	2720	2221
Number of 4 input LUTs	5045	4090
Number of bonded IOBs	129	126
Max frequency	170.180MHz	178.984MHz
Power	784 mW	546 mW
Memory usage for synthesis	245MB	223 MB

V. CONCLUSION

In this paper clock gating technique is applied to optimize the power Consumption of overall architecture (ASP) employing RISC architecture in a single chip. The advantage of proposed architecture is data transferred between the main processing unit and peripheral device increases, the more number of interrupts from various peripheral (External) devices also increases. The designed Architecture power saving up to 30.36%; when processor operated at 1.2v.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.

REFERNCES

[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.Vinaya Babu, “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] Kamaraju.M, Lal Kishore.K, Tilak.A.V.N "A Novel Implementation of Application specific Instruction – set processor (ASIP) using verilog" Waste Issue 59,Nov 2011

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

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

Authors details:

Mr. Shaik Chinna Baji pursuing M. Tech (VLSI-SD) II year in Nimra College of Engineering & Technology from JNTUK.He had his graduation from Prasad V Potluri Siddhartha Institute of technology JNTUK, with branch Electronics & Communication Engineering.

Project Guide:

Mr.K. Tirumala Rao Received the M. Tech degree in VLSI SYSTEM DESIGN from Avanthi Institute of Engineering and Technology, Narsipatnam, B. Tech degree in Electronics and communication Engineering at Nimra College of engineering and Technology. He has total Teaching Experience (UG and PG) of 3 years. He has guided and co-guided 3 P.G

CO-GUIDE:

Mr.M.Muzammil Parvez Received the M. Tech degree in Communication Systems From B.S.Abdur Rahman University Chennai and Currently Pursuing his Ph.D From Bharath University Chennai,.He has total Teaching Experience (UG and PG) of 3 and half years & he has guided and co-guided 5 P.G Students for Projects