ISSN: 2278 – 909X
International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE)
Volume 5, Issue 7, July 2016

TASK SCHEDULING IN MULTICORE SYSTEM ON CHIP

ASHWINI, DR. BASWARAJ GADGAY, SUMAN B. PUJARI

***ABSTRACT*- Multicore microcontroller is most far and widely used system in the real time environment. The efficiency of the multi-task scheduling is also a more important factor and demanded factor. The multicore system contains multiple processing cores, which provides the result at a same time without producing the any delay. The scheduling method consider for the operation is a round robin scheduling, in RR scheduling it provide equal CPU time to all the cores for executing assigned operation.**

The task scheduling will be done on the Xilinx software. It provide the synthesis result and simulation will be carried out in the Modelsim 10.1d software. The various tasks has executed on this software, those are arithmetic, shift, logical and rotate operations.

***Index terms* – MPSOC, Round robin scheduling.**

I. INTRODUCTION

Simultaneous execution of numerous tasks so called multi task, it is very indispensable to provide scheduling for multi task. Scheduling for multi task becomes majorly important for multicore processor, which can be applied to the control oriented applications. Those kinds of applications are mostly applicable in the real time execution of any embedded systems. There are several structures included by tasks in a real time scheme; those are mutex, and synchronization. Task scheduling must encounter some of the requirements such as performance need to be in real time, power consumption essentially very low, and it should meet harsh timing controls. In a multicore system the main intention of task scheduling is to achieve some of the criteria such as completion time of task should be minimized, improving performance in real time, achieving improved load balance is very important in task scheduling.

All these scheduling are not only related to the scheduling algorithm but equally related to scheduling model are very multifarious. Some of the studies have examined about the performance, which should be in real time with some limits. Some research result will be carried out based on memory constraints and energy aware and so on. Based on the relations tasks will be segregated in the real time structures such as synchronization, executing order and communication. Different priorities will be apportioned Accordingly to the relation, for this reason priority based method becomes very necessary and popular.

Multiple processors are used in multiprocessor SOC, these multiple processors usually fabricated in single substrate so it is called MPSOC. These are usually aimed for embedded applications. It is used by the place that contains multiple processing elements with precise functionality reproducing the need of estimated application domain. A two or more cores or processing units are included in multicore which is a single computing section. Those cores can be used to write and read the instructions detailed by the processors. Here the instructions will be like logical instructions, arithmetic instructions, rotate instructions, shift instructions, branch, add and move data, these are the some instructions specified by the processors.

The main advantage of multiple core are it can be able to run multiple instructions at a time which inter provide the increment in speed for the internal program execution so as result it will provide the parallel execution. Here the designer will fabricate multiple cores on single die. Caches can be shared by the cores may be or may not be and they can be able to implement shared memory for interconnected cores communication or message passing. Bus, two dimensional mesh, ring and crossbar are the network topologies which provide the interconnections between the cores.

Similar cores are included in multicore organization which is homogeneous and dissimilar cores in the multicore are heterogeneous. Integration of all the electronic systems and components of processor into a one chip is so called the SOC. A single substrate may contain mixed signal, analog, digital and radio frequency these functions are fabricated into single substrate.

There is advantage of very less power consumption of SOC so it is widely used in mobile electronic field. The main area of using the SOC is an implementation of embedded applications. Microcontroller which is also called the single chip devices because it include an in built of 100KB of random access memory. But for desktop version of Linux and window, SOC will be act as running software so it is a capable of handling powerful processors so these kinds of the processors will be used as external memories such as RAM, flash for the chip and also these can be used as external peripherals.

As a high degree of integrating chip, as a result production of very smaller systems and direct reduction on manufacturing cost are always aim of the system on chip. CPS is an integration of physical process, computations and networking. It will be widely used in the application domain

such as chemical process, energy, manufacturing, environment, transportation, consumer appliances, heath care, civil infrastructures, aerospace and automotive.

In some environment SOC is not able to implement some exact applications, so instead of using SOC we can use the SIP. In SIP, number of chips will comprise in a single bundle so it is known as SIP. While making comparison to SIP the SOC is a cost efficient, reliable, and good in performance these all aspects are come into picture when considering the large number of volumes.

The manner in which scheduling activities are carried out is referred as scheduler. The key objective of the scheduler is to maintain all the resources of the computer to be busy and effective utilization of all the computer resources in good manner or achieving the expected quality to the requested services.

The objective of scheduling polices are throughput should be maximized, response time should be minimized, minimized latency and the scheduler should provide equal priority to the all tasks for sharing resources among differently requested tasks.

II. LITERATURE SURVEY

[1] Zhang-Jie[1], The transportation system, manufacturing, power plants and hospitals are the reliable systems in which it is very necessary that task apportioned to the system will be done within the billed time otherwise it will became a source for disastrous significances. The correctness of any system will not only depends on the scheming of accurate logic but it is equally depends on overall execution time required for any task. For any RTOS, schedulability test and scheduling algorithm are the heart of the system. When the entire task performs its execution will be in allocated time period then the scheduling algorithm will be considered feasible. In UC/OS-II and Vx works are the operating systems and these OS uses the fixed priority algorithm as a scheduling algorithm.

In this paper author explained about dynamic scheduling method for real time responsibilities. According to previous work earliest dead line must always select the latest task to run, so it is very necessary in EDF that include deadline time and release time in each PCB of system. And also ordering of each deadline from next to extreme, and then arranging all the ready PCB in chain. Another scheduling algorithm in this paper is a priority inheritance protocol. For each lock object in system it is necessary to include the current process blocked by an exclusive lock and current process occupying an exclusive lock.

The author describes the cache ways can be altered via hardware outline register. The partitioned total cache ways will be in the range power of two. And another author Albonesi et al. proposes cache architecture can be capable to arrange new style of configuration to have any number of cache ways. Rendering to the behaviour of task, cache can be reconfigured dynamically to obtain the improved energy efficiency. To progress the performance of real time system cache partitioning is studied.

Task utilization can be minimized by cache partitioning architecture, here this architecture will consider by taking into account task criticality. By influencing configurable cache architecture the author will propose a method to reduce cache energy consumption and eliminate inter task cache interference.

To decrease the cache energy feeding for soft real time system the author proposed another technique. In this literature survey, we propose a method for a reduction of power and energy dissipation for distributed system, multicore process and single processor based on scheduling algorithm, allocation, system and software. Application scheduling algorithm and power management technique are the further achievements that deal with power and energy efficiency.

For multicore system to minimize the overall WCET co-operative consideration of cache partitioning and task assignment technique will be used. The precise WCET can be obtained from cache locking. The core level will contain the portioned shared cache [1].

[2] Wan Yeon[2] for minimizing the overall power consumption of processors numerous investigations on real time tasks had carried out and for each task need to meet their time limit. Most of the studies had based on the overabundant cores and fewer cores. These approaches had been supported for energy saving technique. But the author describes this algorithm with the consideration of taking lightly loaded multicores.

These LLM contains the criteria that should be in manner such that the number of running tasks will be less than the number of processing cores. And most importantly the author explained about minimizing energy consumption of overabundant cores of lightly loaded processors for real time tasks. Minimum energy feasible scheduling can be obtained from polynomial-time scheduling scheme. Wan yeon described two scheduling approaches those are static approach and dynamic approach. In the SA initially all the scheduling decisions are made early for the execution of tasks at offline and in dynamic approach, static approach scheduling decisions will be adjust during runtime or online. So here the overall calculation time for the task will be reduce as compared to the static approach.

The main motivation form this paper will provide two main good reasons first one is to provide the parallel processing for overabundant cores for achieving the energy saving capabilities. Because here the tasks will be divided into many subtasks and these subtasks will be completes their works on the multiple different cores due to all these reasons there is a rapid minimization in the execution time with lowest frequency. From this it is clear that executing any task on single core will take more time but executing similar task on multicore requires the less time for their task execution.

The second motivated solid reason is to save a power for infrequently used cores. Once the processor will allot task to the cores, then processor will switch off the power of not frequently used cores. Optimal frequency is known as executing all the processor allotted tasks within the time limit. Here considering the 'N' similar (homogeneous) processing cores and all these cores need to operate at same time with different clock frequency. Not used cores need to change their mode with a dormant mode.

The energy consumption of dormant mode will be negligible and dormant mode will be also referred as power off. If the task work will be completed before given time, then those tasks will be put into power off mode or dormant mode. The cores can be possible to adopt the pre-fetch technique because this can will be know before the next instruction and data before, which need for the execution of faster operation so it possible to hide the memory access latency.

If consider the above case then, from the on chip clock frequency it is possible to determine operation speed of single core used in the entire system. From this paper the author explained two rich techniques for power saving. One for overabundant cores, for these cores by applying parallel execution of task on those cores will be possibility to reduce the power ingestion. And the second rich technique is the powering off the hardly ever used cores [2].

III. PROPOSED METHOD

3.1 DESIGN METHODOLOGY

There are so many weaknesses of the conventional approach those are, in the conventional design methodology will make use of four clock cycles and requires the clock for each core request to transmit or receive the data, exploit more number of gates, the power consumption will be more because it requires total four clock cycles to place a communication between one processor to another, and it requires the arbitrary memory to provide the communication between all the processor. For arbitrary memory separate logic circuit is necessary to be designed and every processor in the existing technology will have its own dedicated memory.

To overcome the downsides of convention design style we propose the task scheduling architecture for quad core processor is designed. In this proposed method employs less number of gates, the overall power consumption compare to the existing technology will be less due to this there is improvement in the performance and latency will also be less. Architecture of shared memory quad core processor is shown figure 4.1.

Below figure 4.1 shows the design methodology of proposed design. Design consist of many blocks those are decoder, FIFO, shared memory, task scheduling (round robin scheduling) and four numbers of cores.

3.1.1 FUNCTIONS

Harvard architecture, Pipeline execution.

Processor consists of:

- Fetch instruction
- Decode Instruction
- Execution Instruction
- Register(W/R)

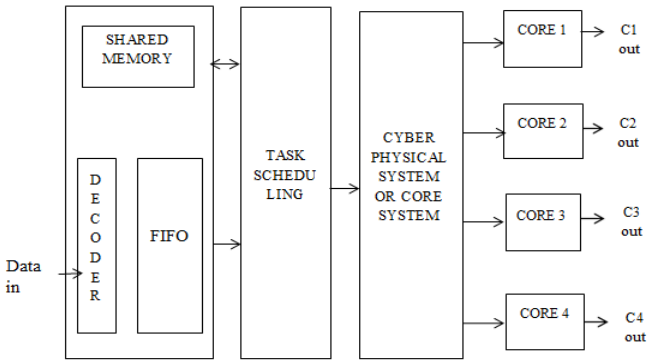

Fig. 1 Task Scheduling Architecture with Quad-Core Processor

3.1.2 DECODER:

Set of signals are converted by code is named as a decoder. There are so many applications for decoder those are containing memory address decoding, data de-multiplexing and seven segment displays. There are so many types of binary decoder, but when we consider decoder for all the application, decoder consists of multiple inputs and multiple outputs and decoder is an electronic circuit.

Decoder will be implemented as a part of more complex IC (integrated circuits) or it will be implemented as stand-alone IC. The HDL such as VHDL and Verilog are used to synthesize the decoder function. In an advice methodology the decoder block will receives total 21 bit of data, initially for design methodology it's unaware of data, in the sense which was the opcode and the total 16 bit of inputs data. Here for the proper operation it is necessary to know opcode and the required input pattern.

So decoder will perform identification of data to decide which was the opcode pattern and which is the data input, so it consider first 5 bit out of 21 bit as opcode and the remaining 16 bit as data input bit. And it will transfer total 21 bit of data to the FIFO.

3.1.3 FIFO:

FIFO is total 21 bit of input and output port. Input port of the FIFO is controlled by the write enable pin and free running clock and output port of the FIFO by read enable pin. When the write enable pin is asserted data is written into the FIFO and when the read enable pin is emphasized data is read out from the FIFO.

To indicate the state of the FIFO there are two flags used those are FIFO empty and FIFO full. The FIFO empty flag goes high when there is no data present in the FIFO and the FIFO full flag energies high when all the data transferred from the decoder will be present in the FIFO or there is an availability of data in the FIFO. FIFO is manufactured using high speed submicron CMOS. "Stuck at fault" are another common problem in the memory, the effective way out to this problem is the error correction and error detection.

FIFO will receive the total 21 bit of data input and it transfer LSB of 8 bit as an input 'A', MSB bit as input 'B' and first 5 bit out of 21 bit as opcode. Here opcode will be represents the various operations to be performed like logical operations; arithmetic operations; rotate and shift operations.

In the figure it is shown that each part of the block diagram will take the values of data input and its corresponding outputs. There are many blocks present in the diagram in which the first block will be the decoder which splits total 21 bit of a data into single 5 bit of one data and other is 16 bit of data.

Then corresponding result of the decoder will be passing on to the FIFO. And the total depth of FIFO will be the two to the power of 21. Again the FIFO will separates the total 16 bit of data into two 8 bit of data. One 8 bit data will be considered as input 'A' and another 8 bit of data to be seeing as an input 'B'. Then by using shared memory, data will be transmitted to the processor for handing over the works to their cores.

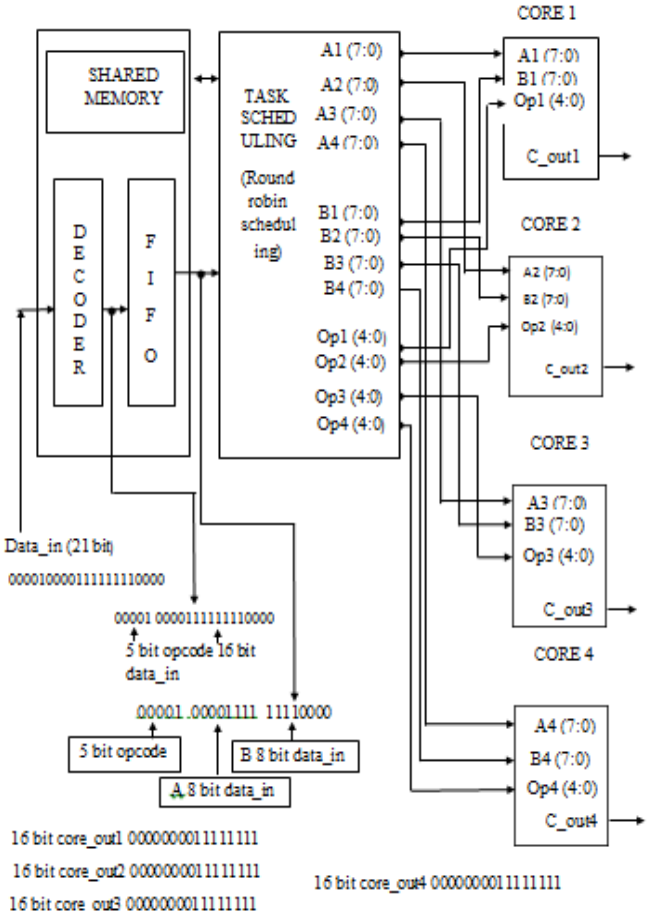

Fig. 2 Detail block diagram of task scheduling algorithm for modern CPS

3.1.4 SHARED MEMORY:

A memory is shared memory that may accessed by multiple programs to provide the communication among the various programs or to avoid unnecessary copies. The efficient way of transferring a data between the programs termed as the shared memory. Depending on context, program may run on multiple separate processors or programs may run on single processors.

Shared memory provides the facility that processors can write/read from memory at the same time. Here priority method will be considered when write address collisions are occurred.

Shared memory encompasses of:

- Data output/input.
- Write/read operation.
- One clock.
- Address.

Shared memory design:

Machine frequently supports both the private data and Shared data in the shared memory. Single processor will make use of private data in a shared memory, while multiple processors will make use of shared data which holds to provide the communication surrounded by the various numerous processors at unchanged time and communication will be in the form of reads and writes of the shared data.

When the private data is reserved, private data location transferred to the cache location, minimizing the memory bandwidth requirement in addition to average access time. As name itself implies private data is made for single processor. So its behavior is identical to the uniprocessor.

When the shared item is reserved, the value stored in shared data will be replicate into multiple caches in addition to that; there is a reduction in required memory bandwidth and access latency.

Basic figure of centralized SMM (Shared memory multiprocessor) is shown in below figure 4.3

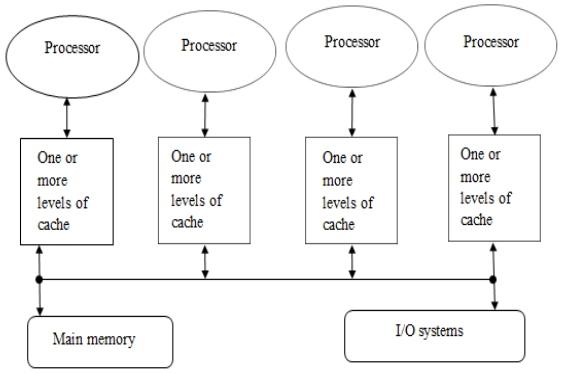

Fig. 3 Basic structure of a centralized shared-memory multiprocessor

A same physical memory is shared by multiple processor cache sub systems. Typically those are connected to one or more switches or buses. The main aim of this architecture is the equal access time to all of the memory from all the processor.

3.1.5 TASK SCHEDULING PROCEDURES

Distributing resources between the parties which asynchronously and simultaneously request them it is referred as scheduling disciplines and sometime scheduling disciplines are also named as algorithms. These scheduling disciplines will be used in operating systems for sharing the CPU time for both the processes and thread as well as it is used in routers for handling the packet traffic and it is likewise used in printer for printing spooler and disk driver for output and input scheduling etc.

The main objective of scheduling is to decrease the resource starvation. Meaning that it is problem where the process will not get essential resources to complete its task. This type of problem will mainly occur in concurrent computing due to starvation it has chances to create error in scheduling. And another objective of scheduling is to

distribute resources between all the requested parties equally.

The main problem associated with the scheduling is to decide which request to be allocating the required resources. There are many scheduling algorithms will be present those are listed below. Before discussing the various scheduling methods first discuss how the core will select the required availability of tasks and the availability of cores.

Now considering the entire task will be partitioned and for a partitioned task the priorities will be fixed. To achieve a better scheduling method the following consideration will be considered.

1. In a ready list all tasks will be in ready state.
2. Some of the scheduling characteristics need to be satisfied by each task, some of the features can be pronounced as T ($C_{num,}$ C_{flag}, and P). Core number will be designated by C_{num}, weather the task is executed will be pointed as C_{flag} and the task priority will be P. if C_{flag} will be 1 then task will be running on C_{num} core. If the C_{flag} is 0 then C_{num} will be invalid and task should be scheduled at succeeding time.
3. T_available (P_0, P_1…P_i) here T_available is to best ever the priorities of the task.
4. C_available (C_0, $C_{1.....}C_i$) here C_available is used to specify that the occurrence of core is available or not. If the core is available then C_available is set to 1 otherwise C_available will be set to 0.

3.1.5.1 ROUND ROBIN SCHEDULING:

For the processes, the round robin method will be considered as the simplest algorithm in the operating system. In a circular order for an each task or process prescribed time slice will be given. And in the RR scheduling all the tasks will have an equal priority for each task. The RR scheduling is starvation free, simple and easy to implement.

The RR scheduling will have the capability of solving the other scheduling problems in the computer network. The computer network will have the data packet scheduling problem. The name of round robin will come from the other field. When some person put a share of something in a business then each and every person will take equal amount of contribution in return.

For a fair scheduling the RR will follow the standard time sharing principle. In a time sharing for each process or task in the system will be allotted an equal amount of time slice, so within the time slice each process need to be executed. If that process was not completed its given task in that predefined time slice then task or process will be pre-empted. Quantum or time slice is defined as in RR scheduling each process will be shipped or transmitted in to a FIFO with a time slice for each process.

The RR scheduling is very effective in time sharing background due to its pre-emptive nature and gave an assurance of even-handed response time for work together process. The main objective problem with this algorithm is the length of the quantum. If the quantum is very short in length then the result in lowering the CPU efficiency. If the quantum is too long then cause unfortunate response time.

ADVANTAGES:

1. In RR there is fairness among each task because CPU time is sliced equal for each task.
2. Ready queue follow the FIFO scheduling, in which newly added task will be placed in the end of the queue.
3. This algorithm will be based on the standard time sharing principle.

Some of the operations need to perform by the cores, these operations are specified processor those are arithmetic, shift, logical and rotate operations. And for these operations the specific operand will be assigned. In this project the 5 bit opcode will be consider but in conventional design 3 bit operand is consider it can perform only 8 different operations.

In a proposed design total 32 different operations can be performed but here we consider on 25 different operations. For these different kinds of operation with their opcode and operands are specified in the below tables and input A and input B both are 8 bit wide. Before show the opcode and their operations format, here consider the total operation format.

Table 1 Operation format

OPERAND	INPUT A	INPUT B
ADD	10101010	11001101
Type of operation	Input data stored in ‘A’ input	Input data saved in ‘B’ input

In the above table operand represents the necessary operation to be performed. And the inputs A and B represent the 16 bit of data operation.

Table 2 List of arithmetic operations

OPCODE	OPERATION NAME	TYPE OF OPERATION
00001	ADD	Addition
00010	SUB	Subtraction
00011	MUL	Multiplication
00100	DIV	Division
00101	INCA	Increment A input
00110	INCB	Increment B input
01000	DCA	Decrement A input
00111	DCB	Decrement B input

Table 3 List of logical operations

OPCODE	OPERAND NAME	TYPE OF OPERATION
01001	AN	AND operation
01010	O	OR operation
01011	NTA	Negation of input A
01100	NTB	Negation of input B
01101	XO	XOR operation
01110	N	NAND operation
01111	NO	NOR operation

Table 4 List of Rotate and Shift operations

OPCODE	OPERAND NAME	TYPE OF OPERATION
10000	SAL	Shift left A input
10001	SBL	Shift left B input
10010	SAR	Shift right A input
10011	SBR	Shift right B input
10100	RLA	Rotate left A input
10101	RLB	Rotate left B input
10110	RRA	Rotate right A input
10111	RRB	Rotate right B input

IV. SIMULATION RESULTS

1. Task = Addition
   Format = 000011100000000111100
   Opcode = 00001
   Input A = 00111100
   Input B = 11000000
   Output = 0000000011111100

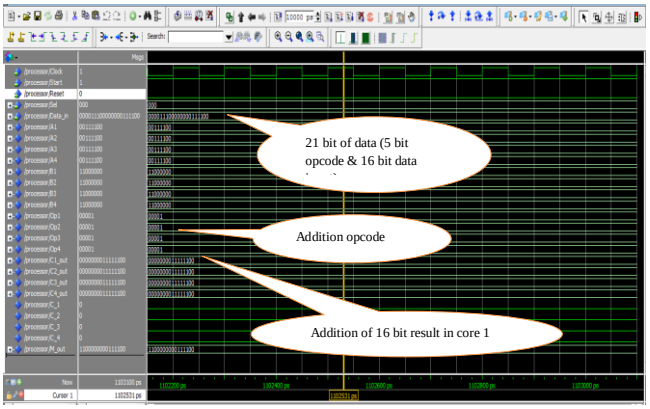

Fig. 4 Simulation result of 16-bit addition

The simulation results will be checked in the ModelSim software. Here in this project various tasks have to be executed. The total execution of task list will be 25 but for simulation results only few tasks have been considered. The first task is addition, in this operation the input pattern is specified that is 00001 11000000 00111100. In this pattern the opcode and input data for performing allocated task operation will be specified. Before getting the simulation results the first and most important work is to adjust some of the things those are first make clock will be equal to clock and reset equal to zero because it need to erase all the garbage values. While performing the simulations then make reset to be one and give the total 21 bit of data input pattern. For given data input it will provide the result based on opcode operation will be specified in the program. The whole program will be written in the Xilinx program and simulation will be carried out in in the ModelSim Altera.

2. Task = Multiplication
   Format = 000111000000010000000
   Opcode = 00011
   Input A = 10000000
   Input B = 10000000
   Output = 0100000000000000

In the third simulation result will be for the multiplication operation for this operation the format will be given by 00011 10000000 10000000. Based on the opcode and inputs the result will be displayed in the all cores. For this operation the total 16 bit of output bit will be utilized because the multiplication of two bit result will be generates the carry result. With respect to all inputs and results will be shown in the result.

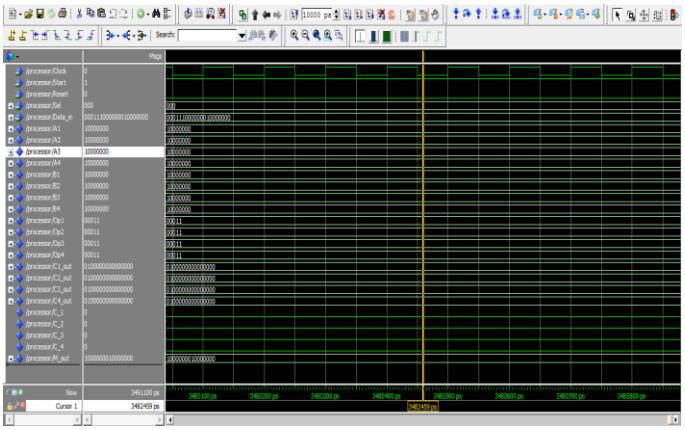

Fig. 5 Simulation result of 16-bit Multiplication

3. Task = SBR
   Format =100111010110000110101
   Opcode = 10011
   Input A = 00110101
   Input B = 10101100
   Output = 00000000U1010110

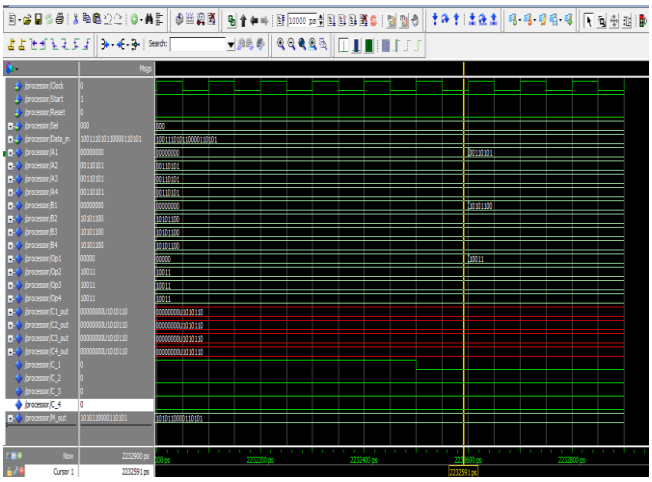

Fig. 6 Simulation result of 16-bit SBR operation

4. Task = All task
   Format = 000000011011010110001
   Opcode = 00000(No opcode for particular task in the program)
   Input A = 10110001
   Input B = 00110110
   Output = the entire task executes; those are stored in the core.

Now if the any opcode is not specified for the operation, then the several set of operations will be specified operations in core will be executes. Here there are various tasks will be assigned. And depending upon the input values all the operations stored in the particular core will be executed. The simulations for this task will be shown below.

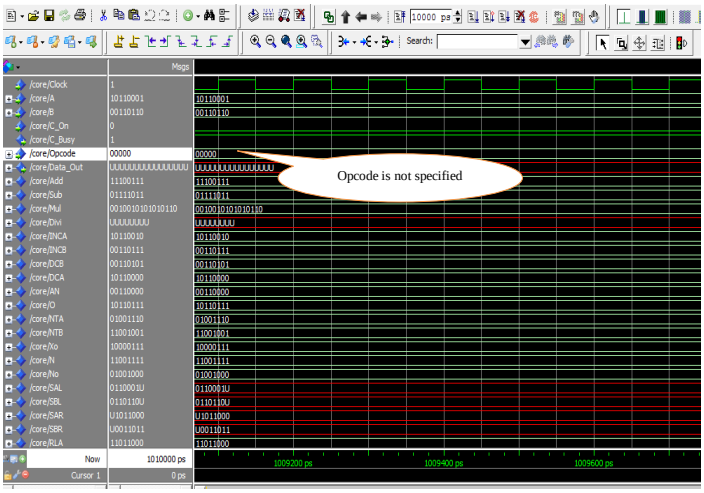

Fig. 7 Simulation a core 1 when there is no opcode is specified

V. CONCLUSION

Due to development in Nano-dimension technology, it allows the implementation of very complex design on single chip so is called the MPSOC. The MPSOC is most needed technology available in this VLSI scale of technology. Due to this technology there is speedy growth in the global networking, autonomous computing, mobile computing and wireless communication.

Task scheduling is done on the multicore processor, a different tasks assigned on these cores with the round robin scheduling policy. Five bit of opcode is considering for selection of the tasks. The multicore processor will produce the parallel results means that without any delay results for the all cores will be obtain. Here the overall concept is that multicore or multi processors, the resources of MPSOC will handle multiple interprets between many users. The communication has been taking place between the cores and the users to provide a result for requested task with great accuracy without any delay as well as performance need to be improved, response time, workload balance, speed up will significantly increase.

In FPGA we will do hardware operations if four users like do some different operations because of using parallel MPSOC the output will be showed without any delay.

FUTURE WORK:

The opcode for the operation and the overall bits can be increases for a wider communication this concept can be used to apply for the wider cyber physical system. In this project on communication will be provided for the cores, because the output of the core can also be making use of input for the further operation. But the future work includes, this concept can be apply to the large communication environment for performing the user assigned tasks.

REFERENCES

[1]. Zhang-Jie "A dynamic scheduling model for real time tasks in reliable system", VOL: 3 IEEE Transaction, pages 185-188, 25-26 April 2009.

[2]. Wan Yeon Lee "Energy efficient scheduling of periodic real time tasks on lightly loaded multicore processors". VOL: 23, Issue: 3, IEEE transection on parallel and distributed systems, pages 530-537, 17 March 2012.

[3]. Jian-Jun Han, Xiaodong Wu, Dakai Zhu, Hai Jin, Laurence T. Yang, and Jean-Luc Gaudiot "Synchronization-Aware Energy Management for VFI-Based Multicore Real-Time Systems", IEEE TRANSACTIONS ON COMPUTERS, VOL: 61, NO. 12, pp: 1682-1695, Dec. 2012.

[4]. Che-Wei Chang "Real time partitioned scheduling on multicore system with local and global memories", page number 467-472 IEEE transection on partitioned system 2013.

[5]. Renato mancuro "Real time cache management framework for multicore architectures" VOL: 61 Issue: 3, Page number 45-54, IEEE transection 9-11 April 2015.

[6]. H. Aydin, V. Devadas, and D. Zhu, "System-Level Energy Management for Periodic Real-Time Tasks," Proc.

IEEE 27th Real-Time Systems Symp. (RTSS '06), pp. 313-322, 2006.

[7]. M. Bertogna, M. Cirinei, and G. Lipar, "Schedulability Analysis of Global Scheduling Algorithms on Multiprocessor Platforms," IEEE Trans. Parallel and Distributed Systems, vol. 20, no. 4, pp. 553- 566, Apr. 2009.

[8]. J. Luo and N.K. Jha, "Power-Efficient Scheduling for Heterogeneous Distributed Real-Time Embedded Systems," IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems, vol. 26, no. 6, pp. 1161-1170, June 2007.

[9] L.Sha, R.Rajkumar, and J.P.Lehoczky, "Priority inheritance protocols: An approach to real-time synchronization" Tech. Rep, Dep. Compute Science, CMU, 1987. IEEE Trans. Compute, vol. 39, pp. 1175-1185, Sept. 1990.

[10]. T. Abdelzaher and C. Lu, "Schedulability Analysis and Utilization Bounds for Highly Scalable Real-Time Services," Proc. IEEE Real-Time Technology and Applications Symp., June 2001

Ms. Ashwini is a PG student studying inDept. of PG studies, M.Tech VLSI Design and Embedded Systems, VTU Regional Centre, Kalaburagi. She has completed her BE in Electronics and Communication from APPA College of Engineering, Kalaburagi.

Dr. Baswaraj Gadgay

Ms. Suman B. Pujari is an Assi Professor in Dept.of PG studies, M.T VLSI Design and Embedded Syst VTU Regional Centre, Kalaburagi. She 8 months of teaching experience and published paper in IJEEE.