International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-8 Issue-2, July 2019

A Design of Intel's IXP Meant for Embedded Systems

Neha Jain, Manoj K Jain

Abstract: ***Technologies to design an embedded system can be of three types: processor technology, IC technology & design technology. Billions of computing systems are built every year for a variety of purpose. They are built within larger electronic devices. These systems perform a one particular function on regular basis. These systems do not recognized by the device's user. These systems are known as embedded system. Broad categories for system implementation are: Application Specific Integrated Circuit, Field Programmable Gate Array, Co-Processor, Application Specific Instruction Processor and General Purpose Processor. From the network processor's designing point of view, it is very important to understand the preliminary characteristics of network applications which are generally based on address lookup, pattern matching, and queuing management which is further classified as Control plane and Data Plane processing.***

Keywords: Network Processor, General Purpose processor, Field programmable gate array.

I. INTRODUCTION

A NP (network processor) is a core heart of network application domain which is actually a programmable IC. It is equivalent to CPU in any computers. The transition from analog to digital in communication industry has motivated development of NP IC. Generally processors can be classified as below.

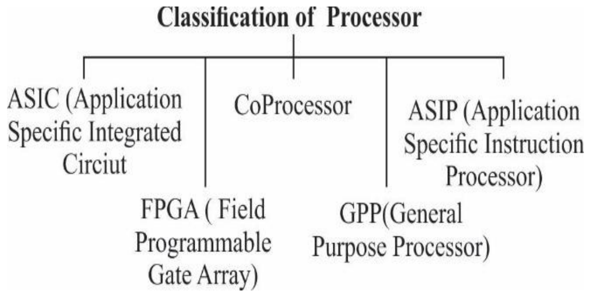

Classification of Processor
ASIC (Application Specific Integrated Circiut
CoProcessor
ASIP (Application Specific Instruction Processor)
FPGA (Field Programmable Gate Array)
GPP(General Purpose Processor)

Fig 1- Classification of Processor

NPs have features of both general purpose processor and ASICs. It gives performance like ASICs and it is flexible and easily programmable like GPP.These are next generation of processors designed for routers and switches. In a typical NP, there are many memory contents which are instructions, control data and packet payload.

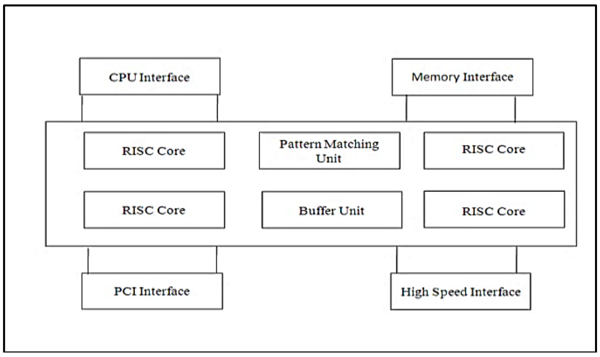


Fig 2- Generic Network Processor

IM i.e. Instruction memory shouldsend concurrent instructions to all the units. As it doesn't possess a large information so it is internally implemented butit's external implementation is possible formore advanced NPwith much more capabilities. Few specific tasks like control processing which takes addresses lookups and it need largerbandwidth for the purpose of its memory. The packet memory need packet processing unit for packet payload processing. For the implementation of those tasks that deal with small data, few processing units are designed for this purpose while other units keep on processing with bulk of information. The classification of memory architecture for NP can be done as shared, distributed and hybrid approaches.

To reduce the load placed on a GPP, Co-processors are used. To perform floating-point calculations, a math coprocessor is preferred. The execution speed of programs performing floating-point operations is dramatically increased after the addition of this coprocessor. Wide varieties of coprocessors are available in market. They can perform very simple such as digital signal processing as well as complex task like crypto graphical functions.

As internet is expanding exponentially, the need for secure communication is increasing very quickly. Thus the performance of application like cryptographic applications and packet processingon network processors becomes an important issue for network processor system design. From network security point of view, Intrusion Detection possess a very important role with Prevention Systems for handling the large internal LAN links.

ECC i.e. Error-correcting codes are popularly used in data storage and digital communication systems to achieve highly reliable storage and data transmission. Due to increasing demand for high-performance and low power systems, efficient hardware design and implementation of ECC become very important.

In one of the application, thisprocessor breaks the incoming video packet into video blocks which are than stored in shared memory.

Revised Manuscript Received on 30 July 2019.

***** Correspondence Author

Manoj Kumar Jain*, Professor in the Department of Computer Science at MohanlalSukhadiya University, Udaipur, India.

Neha Jain, Research Scholar in Department of Computer Science at Mohanlal Sukhadiya University, Udaipur, India.

Retrieval Number: B2873078219/19©BEIESP
DOI: 10.35940/ijrte.B2873.078219
Journal Website: www.ijrte.org

Low pass filtering mechanism is applied to video blocks after their reading by the engines. Later,video blocks are copied to the shared memory. Later, Strong ARM processor repackages these blocks and then it transmits the package. Here, multiple engines work on a single packet, but each engines access a different video block of that packet. Firewall is needed on hardware for quick processing of these packets. Overlay networks are preferred, which are basically connections among end hosts which operate over the top of IP network.Basically the routing operations are performed using user level code, as a result of which, packets undergo much longer delays since the network implementation is done at application level as compared to network level.

II. RELATED WORKS

We have undergone a thorough literature survey of the related work of various processors which is given as below:

a) Application Specific Instruction Set Processors: Gouret al. [1] have proposed that the key step in ASIP synthesis was to decide the architectural feature based on application requirement and constraints. They observed that performance, power and energy consumption was effected by changing register file size.Gloria et al. [2] have proposed some five key mechanisms for ASIP design process. These were architectural design instruction set generation, application analysis, space exploration, hardware synthesis and code synthesis.Fei Sun et al. [3] have proposed that designing of traditional ASICs was expensive and difficult. Due to these drawbacks, manufacturers have started moving towards the programmable solutions. Bhandarkar [4] developed a special class of NP called as Reconfigurable Network Processors (RNP). H.Bos et al. [5] have proposed OS architecture consisting of NP features. M. Bj¨orkman and P. Gunningberg [6] have proposed a model for NP to evaluate various run-time support systems..Hangpei et al.[7] have proposed a novel ASIP design which is meant for FFT computation. Gloria et al.[8] proposedDesign Space Exploration for ASIP. They told that by using extracted parameters from application analysis phase and input constraints, the architectural space was easily explored.Jingnan Yao et al[9] analyzed various types of network processors, based on this analysis, she suggested three types of Network Processors PE topologies. These are: parallel, pipelined and hybrid topology.

b) Network Processor:Byeong et al. [10] have presented a benchmark name- CommBench. It could be used for evaluating and designing of network processors. Tilman [11] proposed the design requirements backbone routers whose throughput satisfied the modern requirements. Their goal was to frame an architecture which could implement network algorithms at the speed of 10 Gb /s to 40 Gb/s.Nemirovsky[12]proposed that many NPs and benchmark were optimized for data plane. Luo et al. [13] have presented NePSim which analyzed and optimized the power dissipation and design of NP architecture.It contained verification framework which automatically perform testing and validation. David et al. [14] have evaluated various industry specific synthesizable processors. They defined three configurations which are based on comparison, performance and area optimization for every processor.Clowley et al. [15] told that to access network (NP4AN), NP was required. Here authors named this processor as access processor. They described main challenges and architectural choices involved in designing of network processors dedicated for access network nodes like DSL Access Multiplexer.Alexanderet al. [16] proposed that NPs play anbig role in the deployment and implementation of the Active Networks. They gave a general introduction of how NPs and ANs were related to each other. Fu and Hagsand[17] have studied current network processors and introduced a network processor model. Niemann et al. [18] have presented architecture for NPs that was based on a uniform, massively parallel structure. They reused predefined IP building blocks. Pai and Govindarajan [19] proposed a framework, called FEADS. Ahmadi et al [20] provided a survey of network processors (NPs). They introduced network processors, described their functionalities, requirements, the architectural specification and implementation of NPs.Cascón et al. [21] proposed and analyzed an offloaded network IPS implementation. For the achievement of good performance, they considered efficient parallel processing for network tasks.Chaurasiya et al [22] proposed that the NP consist of micro engine and processing element.Kanada et al. [23] have told that the development cost of network processor was very high and the number of NP program developers was very less. To solve this problem, authors proposed a language called Phonepl. It was an open, portable and high level programming language. It was designed for the programmer without prior knowledge of hardware and software.

c) IP Lookup and Packet Processing: Shah and Gupta [24] have proposed two algorithms to manage the ternary content-addressable memory so that the incremental update times of the worst case remain small. Kobayashi et al. [25] proposed VLMP i.e.Vertical Logical operation with Mask encoded Prefix length. It is basically an IP forwarding table search engine architecture. Gupta et al. [26] suggested a mechanism to minimize average lookup time. Ruiz-Sánchez et al. [27]presented a survey of the latest and efficient IP address Lookup algorithms. They traced the evolution of the IP addressing architecture.Narlikar and Zane[28] told that previous work has been focused on quantifying memory usageand was aimed at hardware implementations. Taylor et al. [29] have described an efficient and scalable design for lookup engine. According to the author the approach used in high performance systems was custom ASICs and CAM i.e.Content Addressable Memory devices.

d) Queing in network processing system: Fu et al. [30] have studied about the two important aspects, first is the packet delay and the other one is queuing behavior in a NPsystem. They analyzed and modeled the arrival processes for the packet.

e) Deep packet analysis and header classification:Buddhikotet al [31] have presented an algorithm based on space decomposition, called as PACARS i.e. Packet Classification Algorithms using Recursive Space-decompositions PACARS. Woo et al. [32] proposed and studied packet classification which provides good average case performance and according to them it need reasonable storage, and could adapt as per individual filters. Feldmann and Muthukrishnan [33] presented a novel algorithmic framework to solve the problem of packet classification by allowing access times vs. memory space tradeoffs.Baboescu and Varghese [34] proposed a scalable packet classification scheme- ABV i.e.AggregatedBit Vector to propose aggregation and rule rearrangement.

f) Cryptographic algorithm and different types of attacks: Wagner et al. [35] found in security-critical C code, some sort of potential buffer overrun vulnerabilities. According to him, buffer overruns were very common because C was instinctively unsafe. There was no automatically bound checking in array and pointer references. Mirzaee et al [36] have proposed a crypto processor which was based on IDEA (International Data Encryption Algorithm).

g) Signal Processing: Tuck et al. [37] examined string matching algorithms and their use for ITDS. They focused on providing worst-case performance of it. They contributed to modify to the Aho-Corasick string-matching algorithm. Said et al [38] have already proposed the design of an acquisition ASIP and an embedded synchronization which are meant for those systems which are baedon OFDM. Abdelall et al [39]have implemented an orthogonal frequency division multiplexing (OFDM) signal chain processing. Nowadays the OFDM was the most preferred choice for modulation purpose.

Apart from this, we have also analyzed current status of network processors as below:

a. Bay Microsystem’s Implement Chesapeake:As compared to present market scenario, this Chesapeake has the highest performance. It’s a 50Gbps NP which supports traffic management.

b. AU1550® Security NP: Au1550: This one is a low-power, high-performance and high integration security network system which is generally meant for remote access devices such as gateways, NSA i.e.network attached storage units, wireless access points, and VoIP applications apart from Linux based networking.

c. XLP980 NP: Broad Corporation Company has introduced it as the top performance multicore NP architecture.

d. C-5 NP:It’soperating frequencies can be defined as : 166MHz, 200MHz, and 233MHz. It has 32 programmable Serial Data Processors and 17 programmable RISC Cores.

e. CX8620x: In the SOHO and household environments, this is preferred as it provides networking system solutions.

f. EZchipNP:Since it iseconomic in terms of power and space so they are generally preffered in 3G/4G Wireless Infrastructure Equipment, Edge and Core Routers, Data Center Switches and Metro Switches.

g. FP3 NP: This is the first400G network processor with a speed of four times than other contemporay advanced networks with high QoS and tremendous performance.

Apart from these network processors, we have undergone a thorough research and analysis of the industrial well-known brand name intel’s IXP NP as below:

III. INTEL IXP NP:

It is completely programmable micro engines for transmitting of packets and traffic regulation on a chip with 5.4 G operations/seconds. The deep packet inspection is enabled through 2.5 Gbps software pipelining. IXP stands for Internet Exchange Architecture. This NP mainly comprises of three components as : a strong ARM, Microengines (ME) and IXA framework. This strong ARM which is the core processor along with MEs is fully programmable. Here we had undergone thorough research on IXP 1200 which is the first generation of IXP family. Following are the major components of an IXP 1200 Chip

a) Strong ARM core: It’s frequency of operation is 166/200/233 MHz and it’s programmed for control plane applications.

b) Six 32 bit RISC ME: Every ME possess four hardware threads and 8KB programmable instruction control storage which can save upto 2048 instructions.

c) FBI unit & IX Bus: FBI is used for serving fast MAC layer devices on IX bus.

d) SRAM & SDRAM unit: 8MB SRAM to store lookup tables & 256 MB SDRAM for storage of mass data, forwarding information and transmitting queues.

e) PCI unit : It provides 32 bit PCI bus for PCI peripheral devices.

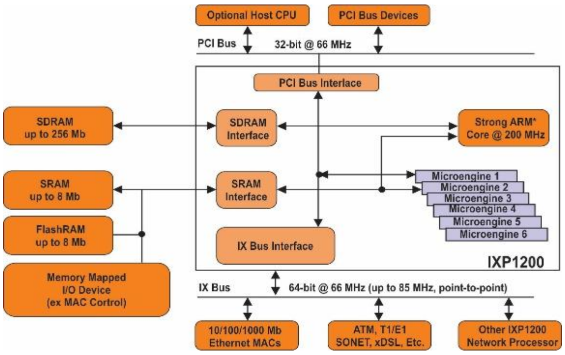


Fig 3 - Intel IXP 1200 Block Diagram.

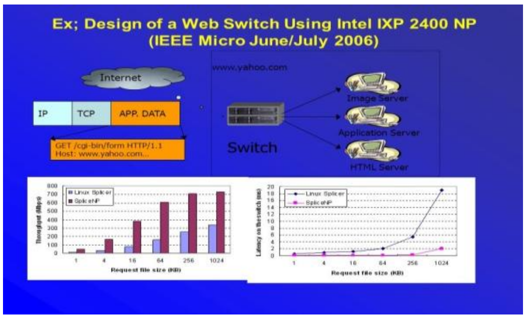

Ex; Design of a Web Switch Using Intel IXP 2400 NP (IEEE Micro June/July 2006)

Fig 4- Application of Intel’s IXP 2400 NP

Retrieval Number: B2873078219/19©BEIESP
DOI: 10.35940/ijrte.B2873.078219
Journal Website: www.ijrte.org

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

Other IXP like IXP 2400, IXP 2800 and IXP 2850 are second generation of IXP NP.

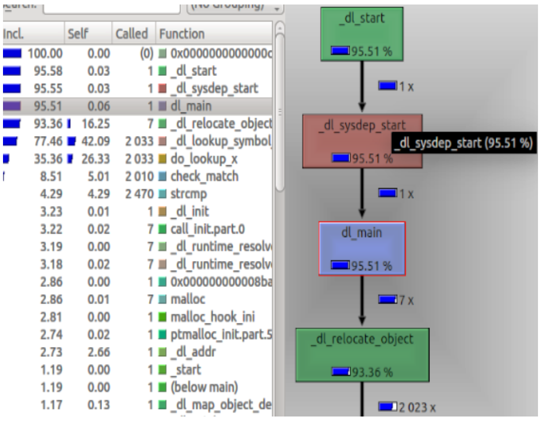
- IXP 2400 is basically designed for 2.5 Gbps (OC-48) applications. It has built in functions for generating pseudo-random numbers and time stamping.
- IXP 2800 is designed for 10 Gbps (OC-192) applications. It has one XScale core processor & 16 MEs.
- IXP2850 is basically IXP 2800 with on chip cryptography units.
- Also the programming languages used by IXP are Microcode which is an assembly language &Microengine C which is a C like language.

IV. PROPOSED METHODOLOGY

Profiling is a way to analyze a program dynamically. It measures the time complexity and space (memory) complexity of a program. It also tells how many times a particular instruction has been used. It tells the duration and frequency of function calls. Mostly profiling information is used for program optimization.

i. Profiling and Analysis of Leaky bucket algorithm: It is used for traffic shaping in data transmissions which means to regulate the data flow in communication channels. It is used to control the data rate in data transmission. Here bucket refers to buffer. Packets are discarded if this bucket is overflow.Packets entered in the buffer at different - different rates with constant output rate. Here, Callgrind profiling tool is used to generate profiling data of few benchmark networking algorithms. When the program runs,Callgrind tool records the call history of all functions and generates a call-graph. It also shows relationship among functions, generates number of execution of an instruction, list of all callers and callees, callees map.For graphical visualization of the data, KDE/Qt based GUI based KCachegrind is preferred. Using this tool it is easy to navigate the large amount of data that is produced by Callgrind.

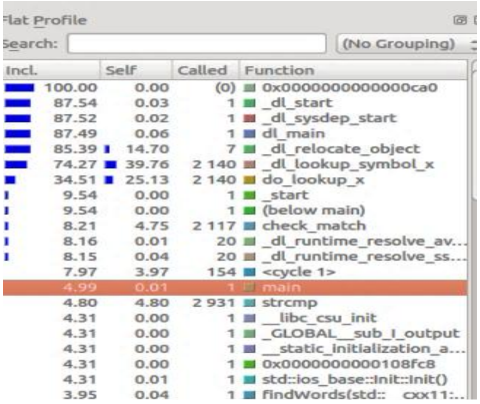
ii. Profiling and Analysis of Borůvka'salgorithm: It is used to find a MST i.e. minimum spanning tree in a graph for which weights of all edges are distinct. Following results shows the profiling data of the Borůvka's algorithm. To generate this data I write the code c++. In fig. 3 we can see that main is not the actual entry point of the program, there are many functions, dynamic link libraries (DLLs) that are executed before and after main. In figure we can clearly see that 0x0000000000000ca0, _dl_start,_dl_sysdep_start,dl_main, dl_relocate_object etc. are the function that are executed before and after main.



Incl.	Self	Called	Function
100.00	0.00	(0)	0x000000000000c
95.58	0.03	1	_dl_start
95.55	0.03	1	_dl_sysdep_start
95.51	0.06	1	dl_main
93.36	16.25	7	_dl_relocate_object
77.46	42.09	2 033	_dl_lookup_symbol
35.36	26.33	2 033	do_lookup_x
8.51	5.01	2 010	check_match
4.29	4.29	2 470	strcmp
3.23	0.01	1	_dl_init
3.22	0.02	7	call_init.part.0
3.19	0.00	7	_dl_runtime_resolv
3.18	0.02	7	_dl_runtime_resolv
2.86	0.00	1	0x00000000008ba
2.86	0.01	7	malloc
2.81	0.00	1	malloc_hook_ini
2.74	0.02	1	ptmalloc_init.part.5
2.73	2.66	1	_dl_addr
1.19	0.00	1	_start
1.19	0.00	1	(below main)
1.17	0.13	1	_dl_map_object_de

Figure 5- Profiling data for dlmain

iii. Profiling and Analysis of Aho-Corasick Algorithm:It's a pattern searching algorithm which can be used to perform ip-lookup, in intrusion detection system etc. In fig. 4 we see that the inclusive cost for main() is 4.99 and the self-cost is 0.01. It means that all the work done in other function. If the inclusive cost is high and self-cost is low we need to optimize the function.



Incl.	Self	Called	Function
100.00	0.00	(0)	0x0000000000000ca0
87.54	0.03	1	_dl_start
87.52	0.02	1	_dl_sysdep_start
87.49	0.06	1	dl_main
85.39	14.70	7	_dl_relocate_object
74.27	39.76	2 140	_dl_lookup_symbol_x
34.51	25.13	2 140	do_lookup_x
9.54	0.00	1	_start
9.54	0.00	1	(below main)
8.21	4.75	2 117	check_match
8.16	0.01	20	_dl_runtime_resolve_av...
8.15	0.04	20	_dl_runtime_resolve_ss...
7.97	3.97	154	<cycle 1>
4.99	0.01	1	main
4.80	4.80	2 931	strcmp
4.31	0.00	1	__libc_csu_init
4.31	0.00	1	_GLOBAL__sub_I_output
4.31	0.00	1	__static_initialization_a...
4.31	0.00	1	0x000000000108fc8
4.31	0.01	1	std::ios_base::Init::Init()
3.95	0.04	1	findWords(std:: cxx11:...

Figure 6- Call Graph for Main)

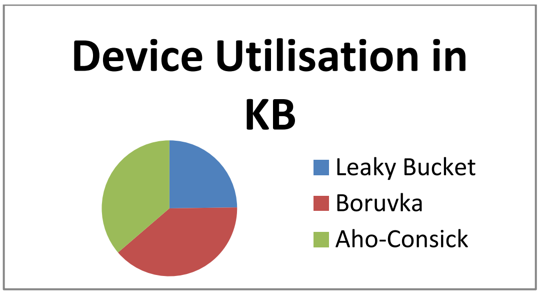
V. RESULTS

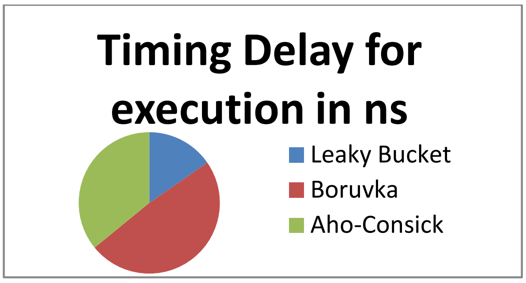
Comparative analysis of selected NPs: Chesapeake NP usespipeline architecture and DRAM Controller. DMA (DDMA) which is descriptor based is used by AU1550. It also supports DES, 3DES, AES, ARC4, SHA1, MD5 while operating in parallel to CPU pipeline.But as compared to others, XLP980 NP is used for network functions like hardware acceleration, virtualizationand deep packet inspection. In C5NP, there are 17 RISC Cores and 32 programmable Serial Data Processors. While CX8620x forms networking system solutions for mainstream application in SOHO and household environments. Last but not the least, forANYBUS NP40™ NP event-based hardware API has been preferred.

Table-1

Parameters	Results of Leaky bucket Algorithm	Results of Borůvka's Algorithms	Results of Aho-Corasick Algorithm
Device Utilization	Total memory usage is in the implementation leaky bucket algorithm is of 197520 kilobytes. Number of slice registers was available 12480 out of which are used 2138. Number of Slice Registers are used 2138 out of 12480 which is 17%. Number of Slice LUTs are used 4129 out of 12480 which is 33%. Number used as Logic is 4129 out of 12480 which is 33%. Numbers of bonded IOBs are used 25 out of 172 which is 14%.	Total memory usage is 310228 kilobytes. Only 2,225 Slice Registers are used out of 55,296 available Slice Registers. Thus only 4% of Slice Registers are used. Similarly only 3,855 4 input LUTs are used out of 55,296 available 4 input LUTs. Thus only 6% 4 input LUTs are used. Thus device utilization is very good.	Total memory usage is 289108 kilobytes. Only 86 Slice Registers are used out of 55,296 available Slice Registers. Thus only 1% of Slice Registers are used. Similarly only 257 4 input LUTs are used out of 55,296 available 4 input LUTs. Thus only 1% 4 input LUTs are used. Thus device utilization is very good.
Timing Delay	The hardware implementation of the algorithm takes only 5.08ns in execution.	Total time consumption is 16.112ns (6.451ns logic, 9.661ns route) (40.0% logic, 60.0% route)	The hardware implementation of the algorithm takes only 11.872ns in all.
Power consumption	Total power supply to the circuit is 0.338 w. Out of which dynamic power supply is 0.017w and quiescent power supply is 0.321w	Total power supply to the circuit is 0.274 w. Out of which dynamic power supply is 0 w so quiescent power supply is 0.274 w	Total power supply to the circuit is 0.274 w. Out of which dynamic power supply is 0 w so quiescent power supply is 0.274 w

Now by the help of following graphical comparison we will get the visualized representation for all these algorithms. So following are the graphs for that purpose.





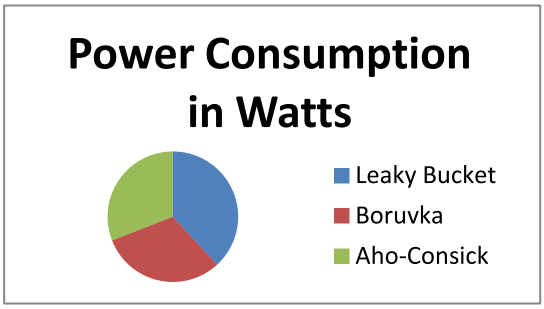
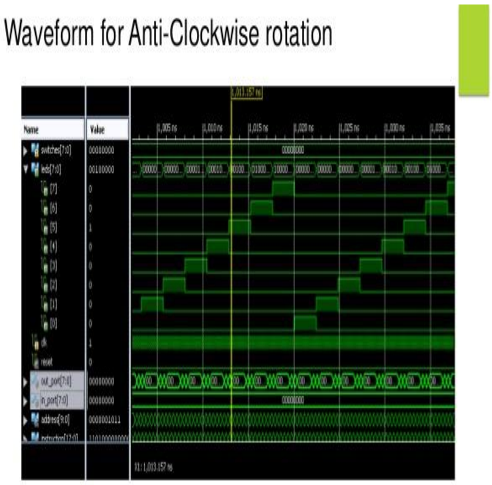
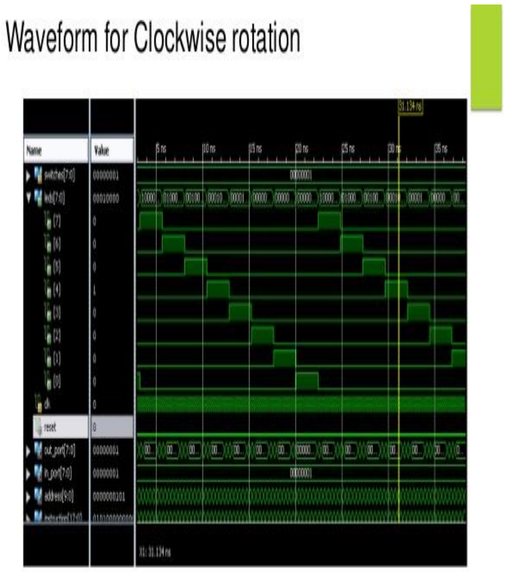


Fig 7: Various parameters ratio

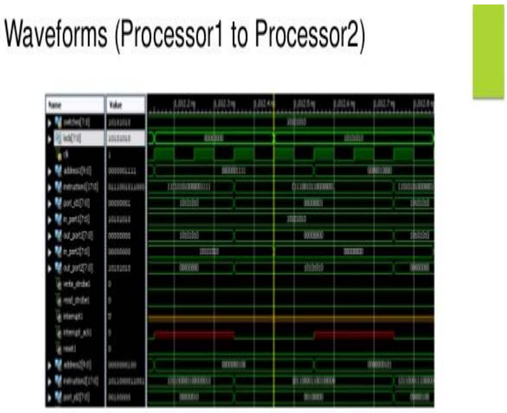


Retrieval Number: B2873078219/19©BEIESP
DOI: 10.35940/ijrte.B2873.078219
Journal Website: www.ijrte.org

Published By:
Blue Eyes Intelligence Engineering & Sciences Publication



Waveform for Clockwise rotation



Waveforms (Processor1 to Processor2)

Switch(0) = 0
⇒ Switches on Processor1
⇒ LEDs on Processor2

Fig 8: Performance analysis

VI. CONCLUSION

This work basically presents a comparative study and analysis of various system implementers for embedded system. Thorough classification of NPs and comparison of algorithms used has been done in this paper. The markets for NPs are tremendously competitive with rapid technological change and up gradation. Presently, there are so many vendors in this industry. Many key players like Intel, Motorola, IBM, Cisco, Broadcom and several other startups are working in design and implementation of Networkprocessors. Currently, Taiwan is the biggest production house for NPs with China Mainland followed the second position. Despite tremendous growth made by various producers, there is a huge scope of betterment as the internet users are increasing every-day. So it is feasible to cater increasing user demand of NP production. The overall global yield of NP is about to reach 339.4 M units by end of this decade and 705.4 M Units by 2022. So it looks that NP manufacturing is there to stay and will grow at a much rapid rate.

REFERENCES

1. Gour,2011," ASIP Design Space Exploration : Survey and Issues ", International Journal of Computer Science and Information Security, Vol 9, No3,2011,pages: 141-145.
2. Gloria" An Evaluation System for Application Specific Architectures. In Proceedings of the 23rd Annual Workshop and Symposium on Microprogramming and Microarchitecture. (Micro 23), pages 80–89, November 1990..
3. Sun "Application-Specific Heterogeneous Multiprocessor Synthesis Using Extensible Processors", IEEE Transactions on Computer Aided Design of Integrated circuits and systems, Vol 25, No.9, Sep 2006
4. Bhandarkar "Performance from architecture: Comparing a RISC and a CISC with similar hardware organization." In *Proceedings of the Fourth International Conference on Architectural Support for Programming Languages and Operating Systems* (Santa Clara, CA, Apr. 8–11). ACM Press, New York, 1991, 310–319.
5. Bos, "FFPF: Fairly Fast Packet Filters". In Proc. of the 6th Symp.on Operating Systems Design and Implementation, pages 347–362, San Francisco, CA, Dec. 2004.
6. Bj¨orkman" Locking effects in multiprocessor implementation of protocols". In Proc. of ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, pages 74–83, Sept. 1993.
7. 7.Hangpei: 'Gaining flexibility and performance of computing using application-specific instructions and reconfigurable architecture'. Int. J. Hybrid Information Technology, April 2009, vol. 2
8. Gloria , "An evaluation system for application specific architectures.", In Proc. Micro-23, 27-29 Nov. 1990, pp. 80-89.
9. Jingnan ―Optimal Network Processor Topologies for Efficient Packet Process Process Processing,‖ IEEE GLOBECOM, 2005.
10. B. Kil Lee; a benchmark suite for control plane and data plane applications for network processors Computer Design, 2003. Proceedings. 21st International Conference on, Vol., Iss., Oct. 13-15, 2003
11. 11.Tilman- A network processor performance and design model with benchmark parameterization. In in Network Processor Workshop in conjunction with Eighth International Symposium on High Performance Computer Architecture (HPCA-8), Cambridge, MA, February 2002.
12. 12.Nemirovsky, "Towards Characterizing Network Processors: Needs and Challenges," XStream Logic (now Clearwater Networks), whitepaper, November 2000.
13. LUO, Y., L. NePSim: A network processor simulator with power evaluation framework. IEEE Micro Special Issue on Network Processors for Future High-End Systems and Applications (Sept/Oct 2004).
14. Patterson and Hennessy (1997), "Computer Organization & Design, second edition", Morgan Kaufmann, ISBN: 1-55860-428-6
15. Clowley, Franklin, and Hamidioglu, Network Processor Design: issues and practices. Morgan Kaufmann, 2003.
16. Alexander, Menage- The price of safety in an active network. Journal of Communications and Networks, 3(1):4–18, Mar. 2001.
17. Hagsand, "Designing and Evaluating Network Processor Applications", In Proc. of 2005 IEEE Workshop on High Performance Switching and Routing (HPSR) Hong Kong, pp. 142-146, September 2005.
18. Niemann, M. Porrmann,"A scalable parallel SoC architecture for network processors", VLSI, 2005. Proceedings. IEEE Computer Society Annual Symposium on, pp.311-313, May 2005.
19. Pai, Govindarajan," FEADS: A Framework for Exploring the Application Design Space on Network Processors", International Journal of Parallel Programming, vol. 35, pp.1-31, February 2007.
20. Ahmadi, "Network Processors: Challenges and Trends", In Proceedings of the 17th Annual Workshop on Circuits, Systems and Signal Processing, ProRisc, pp. 265-269, 2006.
21. Cascón, J. Ortega," Improving IPS by network processors", The Journal of Supercomputing, vol. 57, pp.99-108, July 2011.
22. Chaurasiya, Pandey, "Analysis of Network Processor Processing Elements Topologies", Int. Journal of Advanced Research in Computer Science and Software Engineering (IJARCSSE),vol. 2 pp.66-70, June 2012.
23. Kanada," High-Level Portable Programming Language for Optimized Memory Use of Network Processors", Communications and Network, vol 15, pp.55-69, February 2015.

24. Shah, “Fast incremental updates on Ternary-CAMs for routing lookups and packet classification”, Proceeding Hot Interconnects VIII, IEEE Micro, September 2000.
25. Kobayashi, Kuriyama, “A Longest Prefix Match Search Engine for Multi-Gigabit IP Processing”, Communications, 2000. ICC 2000. 2000 IEEE International Conference on, 2000, pp.1360 -1364, June 2000.
26. Gupta, “Near-optimal routing lookups with bounded worst case performance”, In Proceedings. of IEEE INFOCOM 2000, pp. 1184-1192, August 2000.
27. Ruiz-Sánchez,, “Survey and Taxonomy of IP Address Lookup Algorithms”, IEEE Network, vol. 15, pp. 8-23, May/April 2001.
28. 28.Narlikar “Performance Modeling for Fast IP Lookups”, In Proceedings. of the 2001 ACM SIGMETRICS International conference on Measurement and Modeling of computer systems, vol. 29, pp.1-12, June 2001.
29. Taylor, “Scalable IP Lookup for Programmable Routers”, INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE, pp. 562 - 571, June 2002.
30. J. Fu, O. Hagsand, “Queuing Behavior and Packet Delays in Network Processor Systems”, In Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, 2007. MASCOTS '07. 15th International Symposium on, pp.217-224, October 2007.
31. Buddhikot, “Space decomposition techniques for fast layer-4 switching”, Proceedings. of Conference on Protocols for High Speed Networks, vol. 31, pp. 25-41, 1999.
32. 32.Woo, “A Modular Approach to Packet Classification: Algorithms and Results”, INFOCOM 2000. Nineteenth nnual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE, pp. 1213 – 1222, March 2000.
33. 33.Feldmann, Muthukrishnan, “Tradeoffs for Packet Classification”, INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proc. IEEE, pp. 1193 – 1202, March 2000.
34. 34.Baboescu, Varghese, “Scalable packet classification”, Proceedings. of ACM Sigcomm, vol. 13, pp. 2-14, March 2005.
35. Wagner, “A First Step Towards Automated Detection of Buffer Overrun Vulnerabilities”, In Network and distributed system security, 2005.
36. Mirzaee,” Design and Implementation of an ASIP-Based Crypto Processor for IDEA and SAFER K-64”,International journal of design, analysis and tools for integrated circuits and systems, pp. 21-30, November 2012.
37. Tuck, Sherwood, "Deterministic memory-efficient string matching algorithms for intrusion detection", Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies, pp. 2628 – 2639, March 2004.
38. Said, A. Shalash, “Embedded reconfigurable synchronization acquisition ASIP for a multi-standard OFDM receiver”, EURASIP J. Embedded Syst., pp. 1-16, December 2012.
39. 39.Abdelall, “Design and implementation of application-specific instruction-set processor design for high-throughput multi-standard wirelessorthogonal frequency division multiplexing baseband processor”, IET Circuits, Devices & Systems, vol. 9, pp.191-203, May 2015

AUTHORS PROFILE

Manoj Kumar Jain, Professor in the Department of Computer Science at MohanlalSukhadiya University, Udaipur, India. He has done M Tech. In Computer Applications and Ph.D. in Computer Science and Engineering from IIT Delhi. His area of research includes Application Specific Instruction set Processor or ASIP Synthesis, Embedded System Design, IoT, Computer Architecture, and Hardware Synthesis. He has long experience in research and teaching. He has guided many PhD work in last decade.

Neha Jain. Research Scholar in Department of Computer Science at Mohanlal Sukhadiya University, Udaipur, India. Her area of research includes Application Specific Instruction set Processor or ASIP Synthesis, Embedded System Design. She has long experience in academics & research. She has been a good scholar in her career.