International Journal of Engineering and Technical Research (IJETR)
ISSN: 2321-0869, Volume-1, Issue-2, April 2013

Study of Architectural Design of VLSI

Veni Madhav Sharma, Javed Ali Mansuri, Sunil sharma

Abstract— **The paper presents an overview on architectures for design implementations of VLSI architecture schemes as specified By standardization committees of the ITU and ISO. Implementation strategies are discussed and split into function specific and programmable architectures. As examples for the function oriented approach, alternative architectures will be evaluated. It is also dedicated decoder chips are included. Architectures are presented for reported design examples from the literature. Heterogeneous processors outperform homogeneous processors because of adaptation to the requirements of special subtasks by dedicated modules. Majority of heterogeneous process incorporate dedicated modules for high performance subtasks of high regularity By normalization to a fictive 1.0 pm CMOS process typical linear relationships between silicon area and through-put rate have been determined for the different architectural style. This relationship indicating a figure of merit for silicon efficiency.**

***Index Terms*— Central Processing Unit (CPU), CMOS, SIMD, MISD, Shared Memory (SM).**

I. Introduction

Before going on to the descriptions of the machines themselves, it is very important to consider some mechanisms that are or have been used to increase the performance. The architecture and hardware structure determines to a large extent what the possibilities and impossibilities are in speeding up a computer system beyond the performance of a single CPU. Another important factor which is considered in combination with the hardware is the capability of compilers to generate efficient code to be executed on the given hardware platform. In many cases it is hard to distinguish between hardware and software influences and one has to be careful in the interpretation of results when ascribing certain effects to hardware or software peculiarities. In this chapter we will give most accentuation to the hardware architecture.

This classification is based on the way of manipulating of instruction and data streams and comprises four main architectural points. We will first briefly sketch these classes and afterwards fill in some details when each of the classes is described separately

Manuscript received April 19, 2013

Veni Madhav Sharma : is M.Tech. scholar from Rajasthan Technical University.

Javed Ali Mansuri is Lecturer in Department of Electronics & communication Engineering at Pacific University, Udaipur, Rajasthan, India.

Sunil Sharma is Working as Assistant professor in Electronics & communication Engineering department at Pacific University, Udaipur, Rajasthan, India.

II. Architectural Classes of VLSI design

- **SISD** machines: These are the conventional systems that contain one CPU and hence can accommodate one instruction stream that is executed serially. Now a days many large mainframes may have more than one CPU but each of these execute instruction streams that are unrelated. Therefore, such system should be regarded as (a couple of) SISD machines acting on different type of data spaces. For examples of SISD machines are for instance most workstations like those of DEC, Hewlett-Packard, IBM and SGI. The definition of SISD machines is given here for completeness' sake. We will not discuss this type of machines in this report. **SIMD** machines: Such systems often have a large number of processing units, ranging between 1,024 to 16,384 that all may execute the same instruction on different data in lock-step.

So, a single instruction manipulates many data items in parallel. For examples of SIMD machines in this class are the CPP DAP Gamma II and the Quadrics Ape mille which are not marketed anymore since about 2 years. Never the less, this concept is still interesting and it may be expected that this type of system will come up again or at least as a co-processor in large, heterogeneous HPC systems. Nevertheless, the concept is still interesting and it is recurring these days as a co-processor in HPC systems be it in a somewhat restricted form, for instance, a Graphical Processing Unit (GPU).

Another subclass of the SIMD systems are the vector processors. Vector processors act on arrays of similar data rather than on single data items using specially structured CPUs. When data can be manipulated by these vector units, results can be delivered with a rate of one, two and — in special cases — of three per clock cycle (a clock cycle being defined as the basic internal unit of time for this system). So, vector processors execute on their data in an almost parallel way but only when executing in vector mode. In this case they are minimum times faster than when executing in conventional scalar mode field. For the practical purposes vector processors are therefore mostly regarded as SIMD machines. An example of such a system is for instance SX-9B and the Cray X2.

- **MISD** machines: Theoretically in these types of machines multiple instructions should act on a single stream of data. As yet no practical machine in this class have been constructed nor are such systems easily to conceive. We will disregard them in the following discussions.

- **MIMD** machines: These machines execute several instruction streams in parallel on different data. The difference with the multi-processor SISD machines mentioned above lies in the fact that the instructions and data are related because they represent different parts of the same task to be executed. So, MIMD systems may run many sub-tasks in parallel in order to shorten the time-to-solution for the main task to be executed. There is a large variety of MIMD systems and especially in this class the Flynn taxonomy proves to be not fully adequate for the classification of systems. Systems that behave very differently like a four-processor NEC SX-8 and a thousand processor IBM p690 fall both in this class. In the following we will make another important distinction between classes of systems and treat them accordingly.

- **Shared memory systems**: Shared memory systems have multiple CPUs all of which share the same address space. This means that the knowledge of where data is stored is of no concern to the user as there is only one memory accessed by all CPUs on an equal basis. Shared memory systems can be both SIMD and MIMD. Single-CPU vector processors can be easily regarded as an example of the former, while the multi - CPU models of these machines are examples of the latter. We will sometime use the abbreviations SM-MIMD and SM-SIMD for the two subclasses.

Distributed memory systems: In this case each CPU has its own associated memory. The CPU is connected by other network and may exchange data between their respective memories when required. In contrast to shared memory machines the user must be aware of the location of the data in the local memories and will have to move or distribute these data explicitly when needed. Now, distributed memory systems may be either MIMD or SIMD. The first class of SIMD systems mentioned which operate in lock step; all others have distributed memories associated to the processors. As we see, distributed-memory MIMD systems exhibit a large variety in the topology of their connecting network. The details of this topology are largely hidden from the user which is quite helpful with respect to portability of applications. For this distributed-memory systems we will sometimes use DM-SIMD and DM-MIMD to indicate the two subclasses.

III. WHAT IS DESIGN FLOW

- The Design flow is a Standardized design procedure for designing any of the digital circuit. It Start from the design idea down to the actual implementation.
  This process encompasses many steps like
- Specification
- Synthesis
- Simulation
- Layout
- Testability analysis and many more

Digital Design Process:-

- Since we know that the design complexity increasing rapidly and it also Increases the size and complexity of any digital circuit. So the various CAD tools are essential to reduce this complexity. Too many CAD tools can be choose from the present trend to standardize the design flow.
- The CAD tools can be choose according to the demand of digital circuit.
- It can be divided on the Based on Hardware Description Language (HDL).
- The HDLs provides formats for representing the output of different types design steps. An HDL based CAD tool transforms from its HDL input into a HDL output which contains more hardware information.

- The Behavioral level to register transfer level
- Register transfer level to gate level
- Gate level to transistor level

IV. CONCLUSION

This paper presents an overview on architectures for design implementations of VLSI architecture schemes as specified by standardization committees of the ITU and ISO. Heterogeneous processors outperform homogeneous processors because of adaptation to the requirements of special subtasks by dedicated modules.

REFERENCES

[1] Google book like Zainalabedin Navabi, Richards woods and some other also helps for preparing for this materials.

[2]. R. Feynman, "Quantum Mechanical Computers*", Optc News*, Vol. 11, pp. 11-20, 1985.

[3]. M. Miller, and G. W. Dueck, "Spectral Techniques for Reversible Logic Synthesis", In 6th International Symposium on Representations and Methodology of Future Computing Technologies, pp. 56-62, 2003.

[4]. Md. Saiful Islam, "BSSSN: Bit String Swapping Sorting Network for Reversible Logic Synthesis", *Journal of Computer Science*, Vol. 1, No. 1, pp. 94-99, 2007.