AN EFFICIENT TECHNIQUE FOR IN-ORDER PACKET DELIVERY WITH ADAPTIVE ROUTING ALGORITHMS IN NETWORKS ON CHIP

[1] Mrs. P. Narmatha, I-M.E (Applied Electronics), Excel Engineering College, Komarapalayam, Namakkal District. narmathapradeep@yahoo.com
[2] Mrs. M. N. Vanitha Devi, AP/ECE Excel College of Engineering and Technology, Komarapalayam, Namakkal District. vanithadevinateswaran@gmail.com

ABSTRACT: Network-on-chip (NOC) designs are based on a compromise among latency, power dissipation, or energy, and the balance is usually defined at design time. However, setting all parameters, such as buffer size, at design time can cause either excessive power dissipation (originated by router under utilization), or a higher latency. Network-on-chip (NOCs) has emerged as a promising on-chip interconnects for future multi/many-core architectures as NOCs are able to scale communication links with the growing number of cores. I propose a new adaptive routing algorithm for 8-port router Architecture and the new router topology which is used to reduce the network routing time, and it is a suitable alternate to network design and implementation. Network-On-Chip Multi port Router can be modelled using Verilog HDL and simulated using Modelsim software.

Keywords: Cartesian network, Network-on-chip (NOC), Adaptive architecture, Router.

1 INTRODUCTION

Within the next decade, it will be possible to integrate hundreds of billions of transistors on a single chip, which will allow for the integration of hundreds or even thousands of processor cores (a multi/many-core architecture)on a single die along with the interconnect infrastructure and memory. This type of system is likely to be more communication-centric. The fact that interconnects need special attention in upcoming multi/many-core systems has already been established several years ago when research started to focus on the network-on-chip (NoC) paradigm.

In a multi/many-core architecture, the interconnect occupies large amount of the on-chip area, i.e., a large amount of transistors that otherwise might have been deployed for increasing the number and the complexity of the computational resource snow need to be deployed for designing the communication infrastructure.

Considering the case where multiple applications will be running on the system at different time domains adaptively would be very beneficial since a fixed topology with affixed number of buffers most often covers efficiently certain scenarios and represents an over-design in all other cases. So far, NoCs have been studied as on-chip communication architectures for design time parameterized/application-specific (static) multi/many-core systems.

An application-specific NoC is defined as a design-time parameterized architecture with a fixed routing scheme, a fixed number of allowed virtual connectionists each output port, or a fixed application mapping instance. They are generally tailor-made for a certain applications or application domains.

Extensive research has been done to develop area- and energy-efficient NoCs since 2000, both in the industry and in academia. Besides several research breakthrough in developing application-specific NoCs, i.e., Xpipe and Ethereal recently, several general-purpose NoCs such as Tile*64*by Tilera and the 48-coregeneral-purpose Single-chip Cloud Computer (SCC) from Intel have been fabricated.

Design-time parameterized general-purpose NoCs are over-designed considering different types of traffic scenarios and cannot adapt architectural parameters communication resource assignment (lower resource utilization). In a nutshell, the research in the domain of NoC has been mainly concentrated on application-specific NoCs and design-time parameterized general-purpose NoCs of reconfigurable hardware, e.g., field-programmable gate arrays (FPGAs), are not considered in this paper).

NoCs as a system-on-chip communication architecture, its different design methodologies, and key research problems. In, the authors have identified several parameters that need to be researched and parameterized for application-specific NoCs. The key NoC research problems

are in general: customization of the topology and the channel width, buffer sizing, floor planning, routing/switching, scheduling, and IP mapping problems.

2. MODIFIED W-XY ROUTING ALGORITHM

2.1DESIGN OF W-XY ROUTING ALGORTHIM

Network on Chip (NoC) is slowly being accepted as an important paradigm for implementing communication among various cores in a SoC .With the advances in integrated circuits (IC) manufacturing a constant attempt has been to design enormous amounts of networks on the same chip so as to accomplish more resourceful and optimized chips. Further an efficient routing algorithm can be useful to increase the efficiency of the networks embedded on the chips.

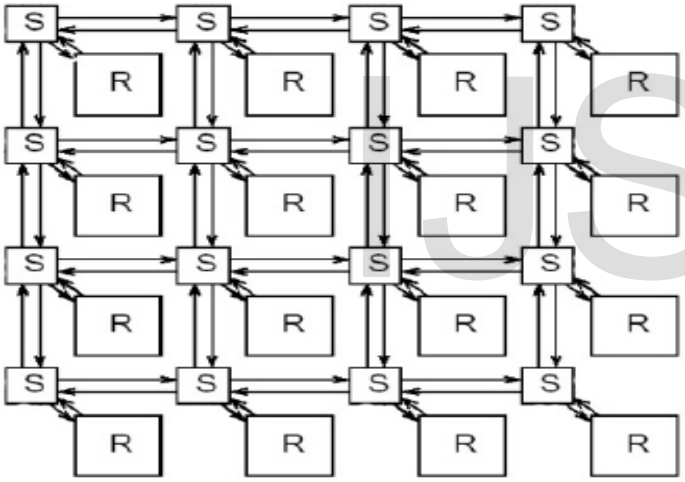

Figure: 1Structure of 4x4 NOC

2.1.1 TYPES OF NOC

The network traffic in NoC network is classified into two types as:
Guaranteed Throughput (GT) Traffic
Best Effort (BE) Traffic

Guaranteed Throughput (GT) also sometimes referred to as Guaranteed Service (GS) works superlatively well with circuit switched network type routing algorithms, where the GT supplier assumes that the sender complies with networks operation requirements.

Best-effort network traffic as the name suggests makes the best effort to route the packets to the destination but there is no guaranty of packet delivery. Since BE is not concerned with delivery of the packet it uses less overheads as compared to GT above. In such cases latencies can be more variable and in the worst situations packets can be lost.

2.2 NOC ROUTING BASICS

In networks on chip (NoC), routing is done to connect different components together in the most efficient manner

2.2.1 NOC ALGORITHM

These algorithms are often classified in two groups.

- Deterministic algorithms
- Adaptive algorithms

The deterministic algorithms usually have less computation and are able to find the next hop in a shorter time. The deterministic algorithms are mostly based on indexes and therefore are more suitable in NOCs which have predetermined structures.

2.2.2PROPOSED METHOD

The proposed method uses a modified XY routing algorithm combined with a scheduler to be used on NoCs. The proposed method uses the 48 bit packet.

XXX	1	DATA	DESTINATION ADDRESS	SOURCE ADDRESS

Table:1 packet network routing

First three bits are don't care, request bit is initially set and is used by the scheduler. Then 32 bit data, the last 6 bits are the destination address and the destination address. For Routing, the scheduler will set the required bit for these nodes and will further schedule these nodes to send data one by one based.

Packets with Req bit set are only forwarded across the network. All packet having Req bit is undergoes for the XY routing algorithm depends upon the priority given by arbiter. Packets are store in FIFO unstill its get grand signal from arbiter. Once get grand signal actual XY Routing takes place.

2.3 SIMULATION RESULTS & ANALYSIS

The simulation is performed on ModelSim SE PLUS 6.2c Simulator. The packet size is 48bits.

The simulation runs on clock frequency of 8 GHz. Synthetic traffic generators generate traffic in the first 2 clock cycles with warm-up period of 1 clock cycles. Performance metrics include average latency per bit and average throughput packets per second for each channel. Packets are generated at each node according to a Gaussian processer. At a time minimum 1 and maximum 9 packets have been delivered. Traffic distribution pattern is uniform 3X3 mesh topology is used for the simulation.

2.3.2 DIRECTION OF PACKET MODULE

The router has four ports (West, South, East and North) to connect with other routers and a local port to connect with PE means have five input and five outputs, as shown in figure 1. Router basically consist of Routing algorithm and switching techniques, where routing algorithm proposed selection strategy which based on the concept of neighbours-on-path, The routing algorithm can be used in order to avoid faulty or congesting ports.

Switching techniques determine when and how internal switches connect their inputs to outputs and the time at which message components may be transferred along these path .The overall performance of a NoC depends on several network characteristics, such as topology, routing algorithm, flow control, and switching technique.

3 RECONFIGURABLE COMPUTING

3.1 TECHNOLOGY

Reconfigurable computing as a concept has been in existence for quite some time [Estrin et al. 1963]. Even general-purpose processors use some of the same basified as, such as reusing computational components for independent computations, and using multiplexers to control the routing between these components. However, the term reconfigurable computing, as it is used in current research (and within this survey), refers to systems in corporation some form of hardware programmability—customizing how the hardware is used using a number routing connection (right).

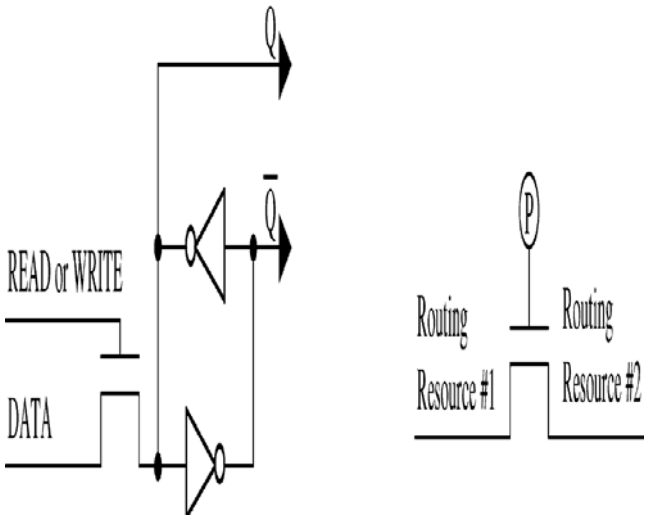

Figure: 2A programming bit for SRAM-based FPGAs [Xilinx 1994] (left) and a programmable routing connection

3.2 PROGRAMMABLE ARRAY LOGIC

These control points can then be changed periodically in order to execute different applications using the same hardware. The recent advances in reconfigurable computing are for the most part derived from the technologies developed for FPGAs in the mid-1980s.

FPGAs were originally created to serve as a hybrid device between PALs and Mask-Programmable Gate Arrays (MPGAs).Like PALs, FPGAs are fully electrically programmable, meaning that the physical design costs are amortized over multiple application circuit implementations and the hardware can be customized nearly instantaneously.

3.2.1 STATIC RAM

Thus, these chips can be programmed and reprogrammed about as easily as a standard static RAM. In fact, one research project, the PAM project [Vuillemin et al.1996], considers a group of one or more FPGAs to be a RAM unit that performs computation between the memory write(sending the configuration information and input data) and memory read (reading the results of the computation). This leads some to use the term Programmable Active Memory or PAM.

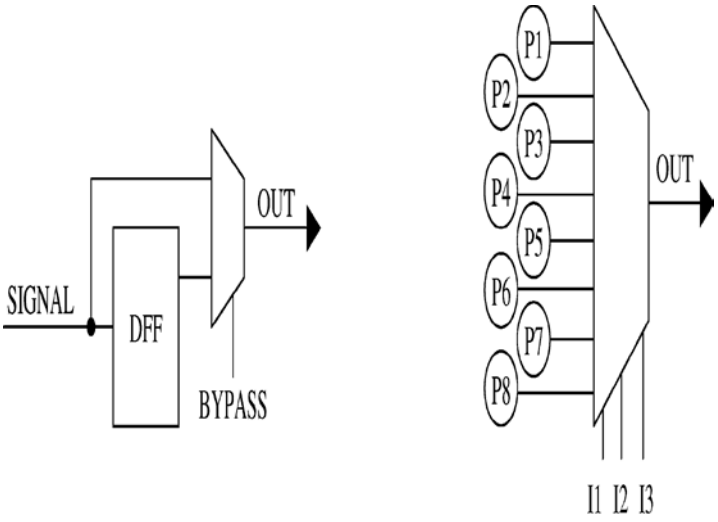

Figure:3 D flip-flop with optional bypass (left) and a 3-input LUT (right).

Finally, the configuration bits may be used as control signals for a computational unit or as the basis for computation itself. As a control signal, a configuration bit may determine whether an ALU performs an addition, subtraction, or other logic computations. On the other hand, with a structure such as a lookup table (LUT), the configuration bits themselves form the result of the computation (see Figure 2 right). These elements are essentially small memories provided for computing arbitrary logic functions. LUTs can compute any function of *N* inputs (where *N* is the number of control signals for the LUT's multiplexer) by programming the2*N* programming bits with the truth table of the desired function. Thus, if all programming bits except the one corresponding to the input pattern 111 were set to zero a 3-input LUT would act as a3-input AND gate, while programming it with all ones except in 000 would computer NAND.

3.2.2 FPGA LOGIC GATES

Since the introduction of FPGAs in themid-1980s, there have been many different investigations into what computation element(s) should be built into the array[Rose et al. 1993]. One could consider FPGAs that were created with PAL-like product term arrays, or multiplexer-based functionality, or even basic fixed functions such as simple NAND and XOR gates. In fact, much such architecture has been built.

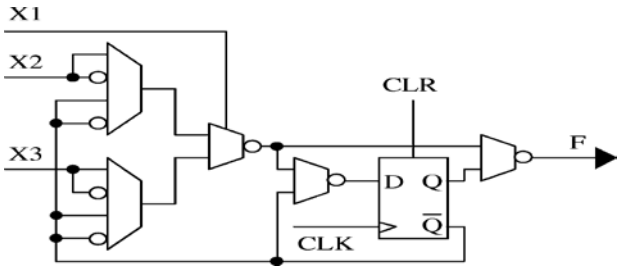

Fig: 5. The functional unit from a Xilinx 6200 cell

This flexibility, with relatively simple routing requirements (each input need only be routed to a single multiplexer control input) turns out to be very powerful for logic implementation. Although it is less area-efficient than fixed logic blocks, such as a standard NAND gate, the truth is that most current FPGAs use less than10% of their chip area for logic, devoting the majority of the silicon real estate for routing resources.

4 VOLTAGE GENERATORS

ABGB is of crucial importance for performing DPM in FinFET-based circuits because it is very effective and easy to implement. While popular techniques like dynamic voltage scaling (DVS) can reduce active-mode power, they incur significant transition latency and energy overhead. ABGB, on the other hand, can switch the back-gate bias voltages of FinFETs very quickly, with little transition energy overhead. In order to enable the ABGB technique for VPSRDPM, we incorporate voltage generators into the buffer banks and the crossbar. The CMOS voltage generator circuit from is adapted to FinFETs for this purpose.

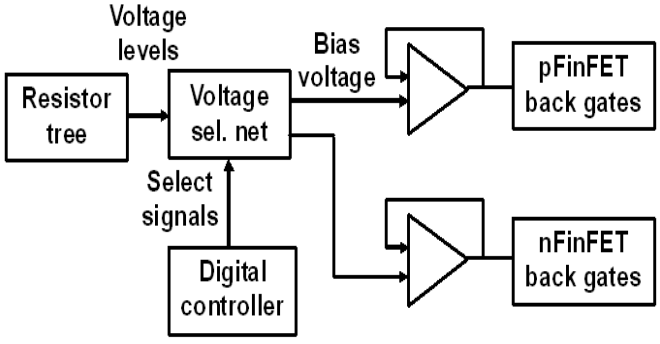

Fig: 6 Voltage generator block diagram

We optimize the design for fast transition (within 1ns or one clock cycle at 1GHz) from normally-biased (*VHI* =1.0*V* and *VLOW* =0*V*) voltage levels to reverse-biased voltage levels (*VHI* =1.2*V* and *VLOW* =−0.2*V*), and vice versa. To accomplish this, the amplifier is scaled to tackle the load

capacitance, which is equal to the sum of the interconnect capacitances and the back-gate capacitances of the FinFETs in the targeted circuit component

4.1RESISTOR TREE

This block consists of a series of resistors. These resistors divide the voltage range (VH to VL) and provide the required reference voltage levels. The resistance values are designed to be high so that the power consumed by the current flowing through it is minimized.

4.1.1VOLTAGE SELECTION NET

The function of the voltage selection net is to select a voltage level from the resistor tree and feed it to the unity-gain amplifier. Parallel pass transistors are used for this purpose, each connected to a tap in the resistor tree, providing various voltage levels for the unity-gain amplifier. Only one pass transistor is turned on at a time.

4.1.2UNITY-GAIN AMPLIFIER

The unity gain amplifier drives the back gates of FinFETs to the reference voltage level received from the voltage selection net. The sizes of transistors used in the amplifier are scaled appropriately to enable this. Two unity gain amplifiers are required: one for p-type FinFETs and the other for n-type FinFETs.

4.1.3DIGITAL CONTROLLER

This unit provides the digital selection signals for the voltage selection net. These signals are determined by the flow control mechanism. Since energy is consumed every time the output of the voltage generator is switched, a corresponding amount of leakage energy should be saved in FinFETs to reach the breakeven point. For VPSR, the breakeven point is only four clock cycles. Our flow control mechanism updates the back gate bias voltages only every 1,000 cycles. Thus, the energy saved by VPSR far outweighs the energy consumed by the voltage generator.

CONCULSON

The proposed AdNoC architecture is the first approach of an adaptive on-chip communication architecture. It provides an adaptive route allocation algorithm to meet varying bandwidth guarantees and an on-demand buffer block assignment scheme for runtime connection establishment between a data producer and a data consumer. The on demand buffer assignment scheme increases the on-chip resource utilization (buffer) and decreases the overall buffer use, on an average, 42% in case study analysis compared to a static approach where a fixed number of buffer blocks are tied to the output port. The entire area overhead (297 slices on an FPGA) is traded off against the flexibility to select available route and on-demand buffer assignment to that route. Ad NoC architecture is complimented by a monitoring component for providing runtime observability. It is hardly intrusive, i.e., in the worst case, it may require a mere 0.7% of the total link capacity. Due to our runtime observability scheme, AdNoC increases the connection success rate by 62% on average compared with state-of-the-art approaches. Furthermore, to provide the first prototype of the AdNoC architecture in FPGA that can efficiently cope with hard-to-predict system behaviour at runtime.

REFERENCES

1. F. Angioliniet al., “Networks on chips: From research to products,” inProc. DAC, 2010, pp. 300–305.
2. L. Beniniet al., “Networks on chips: A new SoC paradigm,” Computer, vol. 35, no. 1, pp. 70–78, 2002.
3. E. Bolotinet al., “QNoC: QoS architecture and design process for network on chip,” J. Syst. Architecture, vol. 50, no. 2–3, pp. 105–128, 2004.
4. S. Borkar, “Design perspectives on 22 nmCMOS and beyond,” in Proc. DAC, 2009, pp. 93–94.
5.L. P. Carloniet al., “Networks-on-chip in emerging interconnect paradigms: Advantages and challenges,” in Proc. NOCS, 2009, pp. 93–102.
6. C. Ciordaset al., “A monitoring-aware network-on-chip design flow,”J. Syst. Architecture, vol. 54, no. 3–4, pp. 397–410, 2008.
7. W. J. Dally et al., “Route packets, not wires: On-chip interconnection networks,” in Proc. DAC, 2001, pp. 684–689.
8.E3S[Online].Available: http://ziyang.eecs.northwestern.edu/dickrp/ e3s/
9.M. A. A. Faruqueet al., “ADAM: Run-time agent-based distributed application mapping for on-chip communication,” in Proc. DAC, 2008, pp. 760–765.
10.M. A. A. Faruqueet al., “QoS-supported on-chip communication for multi-processors,” in Proc. IJPP, 2008, pp. 114–139.

11.C. J. Glass et al., “Maximally fully adaptive routing in 2-D meshes,” in Proc. ICPP, 1992, pp. 101–104.

12 K. Goossenset al., “The Aethereal network on chip after ten years: Goals, evolution, lessons, and future,” in Proc. DAC, 2010, pp. 306–311.

13 P. Gratzet al., “On-chip interconnection networks of the TRIPS chip,”IEEE Micro, vol. 27, no. 5, pp. 41–50, 2007.