Broadcast with mask on a Massively Parallel Processing on a Chip

Hana Krichene
LIFL - INRIA Lille Nord Europe labs
University of Lille 1
Lille, France
National School of Engineers of Sfax
CES lab
University of Sfax
Sfax, Tunisia
Email:hana.krichene@inria.fr

Mouna Baklouti
and
Mohamed Abid
National School of Engineers of Sfax
CES lab
University of Sfax
Sfax, Tunisia
Email: mouna.baklouti@enis.rnu.tn
mohamed.abid@enis.rnu.tn

Philippe Marquet
and
Jean Luc Dekeyser
LIFL - INRIA Lille Nord Europe labs
University of Lille 1
Lille, France
Email: Philippe.Marquet@lifl.fr
jean-luc.dekeyser@lifl.fr

***Abstract*—The delay of instructions broadcast has a significant impact on the performance of Single Instruction Multiple Data (SIMD) architecture. This is especially true for massively parallel processing Systems-on-Chip (mppSoC), where the processing stage and that of setting up the communication mechanism need several clock periods.**

Subnetting is the strategy used to partition a single physical network into more than one smaller logical sub-networks (subnets). This technique better controls the broadcast instructions domain and the data traffic between network nodes. Furthermore, it allows to separate synchronous communications from asynchronous processing which maintains reliable communications and rapid processing through parallel processors.

This paper describes the design of a communication model called broadcast with mask. This model is dedicated to mppSoC architecture with a huge number of processor elements because it maintains performances even when the number of processors increases. Simulation results and an FPGA implementation validate our approach.

I. Introduction

The parallel embedded architectures are increasingly used in the new embedded applications. These applications contain features very demanding in terms of computing power to implement intensive data processing algorithms (image processing, multimedia, computer graphics, etc.). Their growing demand for computing power goes perfectly with the evolution of electronic chips which associate massively parallel architectures with large capacities of integration. To address these complex applications, SIMD architectures are usually used to achieve high performance at low cost. At the hardware level, a SIMD architecture is built up from relatively simple IPs making it easy to adapt the SIMD model to new technologies, such as System-on-Chip. However, the scalability of a SIMD architecture is limited by the broadcast bottleneck [**?**]. In fact, as the number of processing elements (PEs) increases, the wires connecting the control unit and the PEs become longer and signals take longer to propagate. Therefore, the broadcast instructions delay increases and by consequence the performance of the system degrades.

In this work, we focus on the broadcast techniques in the massively parallel processing (mpp) SIMD model, especially we emphasize on the mpp System-on-Chip (mppSoC) [**?**] which is composed of huge numbers of processing elements (PEs) mastered by a particular processor called Array Controller Unit (ACU). The objective of this paper is to overcome the problem of broadcast bottleneck in mppSoC architecture and its influence on the system performance. We propose a model of broadcast with mask (or subnetting) model [**?**] that separates the communication control from the processing. This model consists of segmenting a network into different small networks to adapt the configuration that we want according to the needs. In this case, the mask is a separator between active processing units and inactive ones in our system. The use and control of masks will optimize the operation of our network and separate the sensitive nodes from the others which reduce congestion and predict the evolution of our network.

This paper is structured as follows. Section 2 presents several broadcast techniques in SIMD architectures. Section 3 details the proposed broadcast with mask and focuses on the mask/broadcast instructions. Section 4 discusses the experimental results on subnetting model. Finally, section 5 summarizes the contribution with outlook on future work.

II. Related works

As mentioned earlier, long wires for global instruction broadcast in massively parallel architecture directly limit clock rates [**?**], [**?**]. Many designs attempt to address this problem with limited success. A board level instruction issue rate of 100MHz was achieved in 1995 by Bolotskis Abacus [**?**] and a chip level issue rate of 200MHz is achieved in products from PixelFusion which are developed with the clock rate of 400MHz for 256 PEs [**?**]. This instruction bottleneck limits the scalability of SIMD architectures [**?**], [**?**], [**?**].

Many alternatives to overcome instruction broadcast bottlenecks are developed. The associative computing (ASC) [**?**] was designed within the Department of Computer Science at Kent State University and has been in use since 1970's

with the introduction of associative SIMD computers such as the STARAN [?] and MPP SIMD computers at Goodyear Aerospace [?]. An ASC processor is a SIMD computer that has additional hardware component, called broadcast/reduction network, to support specific high-speed global operations. The fundamental operation in ASC is the associative search. In an associative search, a search key is broadcast by the Control Unit (CU) and in parallel each SIMD PE looks for that key in its local memory. Based on the results of its search, PEs are partitioned into active PEs (responders) or inactive PEs (non-responders). After several revisions the current prototype is a 50 PEs with pipelined instruction execution and a non-pipelined broadcast/reduction network that supports all operations required by the ASC model.

To further improve performance, a fully pipelined broadcast network was combined with hardware multithreading. The multithread associative SIMD processor (MTASC) [?] prototyped in 2007, with 16 PEs with 1KB of memory per PE, 16 hardware thread contexts, operates at clock speed of 75Mhz. The control unit in this model is a multithreaded scalar processor. It consists of a fetch unit, multiple decode units and a scheduler. This approach hides the overage broadcast latency by pipelined instruction broadcast but requires additional instruction latches. Furthermore, it can only support a few number of parallel instructions. This feature limits the usage of recent processors with complex set of instructions.

To accelerate the graphic processing computation, NVIDIA adopts in 2007 the architecture of SIMD PE array called GeForce 8800 GPU [?] which provides massively parallel execution resources and high memory bandwidth. This GPU is comprised of 128 Stream Processors (SPs) clocked at 575Mhz with 728MB of GPU device memory and a 16KB per block shared memory per Streaming multiprocessors(SMs). Despite its high performance, [?] shows the limits of this GPU through: the data broadcast delay which increases linearly with the size of the transferred data (up to 455ms) and the control flow instructions which significantly degrades the performance of GPU program (about 5.5x when compared with lookup tables in FPGA).

With the evolutions of system design, silicon integration technology and demands of applications on the computing power, a SIMD machine in single chip system integration seems important to consider and verify the feasibility of massively parallel models on chip dedicated to complex applications. Thus, a massively parallel SIMD processing System-on-Chip named mppSoC [?], [?] is defined since 2010. This system is generic, parametric in order to be adapted to the application requirements. The control unit in the MppSoC architecture is MIPS processor in which are added specific instructions such as broadcast instruction. In fact, the broadcast technique in mppSoC is simple one-to-all.

While this model did improve performance over the traditional SIMD architecture, it still suffered from the broadcast bottleneck with a great number of PEs. In fact, when the number of PEs increases, the length of connecting wires increases and therefore each instruction takes one or more clock cycles to be broadcasted (depending on distance), this delay increases the execution time as it has been shown on a parallel image processing application in [?]. Therefore, broadcast with mask in mppSoC architecture can be a solution for this kind of problem by using mapping technologies proposed in recent communication networks.

III. Integration of the broadcast with mask technique in mppSoC architecture

A. mppSoC architecture

The mppSoC architecture is mainly based on the traditional architecture of massively parallel system like the MasPar [?]. MppSoc model is a SIMD massively parallel on chip composed of a 2D grid of processing elements (PEs) and memories. Each PE in the grid is connected to its neighbors via a regular neighborhood network [?]. The whole system is synchronously controlled by a control processor ACU (Array Control Unit) which fetches and decodes program instructions and broadcasts parallel instructions to the PEs. At each clock cycle, each PE executes the same instruction under the direction of the ACU in a lockstep manner.

In this architecture, the PEs, working in parallel, need a specific broadcast model for better manage control information and better synchronize communications. Indeed, efficient broadcast of information is crucial to overall system performance.The first implementation of mppSoC architecture achieves the instructions diffusion through a simple one-to-all broadcast mechanism which expects its limits with a great number of PEs. It is, therefore, necessary to define a simple and efficient model of information transfer from a master to a set of slaves in a rapid way and low power consumption. This concept is called: broadcast with mask.

B. Broadcast with mask

1) Subnetting model: The subnetting is a technique to divide a network into smaller networks. In fact, according to a specific mask plan, the network manager can adapt the map addresses to the number of nodes to create smaller network blocks where the dispatching data flow is optimized. These new blocks will help increase the capacity to manage network traffic and use multiple broadcast domains.

To fit the subnetting model in mppSoC architecture, we make some modifications of the architecture to ensure the approach of synchronous communication in the whole system:

- The Network Control Unit (NCU) (i.e. ACU in mppSoC) and PEs are designed with a 16 bits hardware Forth [?] processor.

  Our model is generic and can support any kind of processors (Mips, Homade, Forth, etc.). For our fist implementation we chose the Forth processor. This choice allows us to use short instructions because Forth is a stack processor which does not need to reference the operands as in processors with registers. It can therefore be used successfully on embedded systems where memory is limited. In addition, access is only possible for the first elements; the circuits can be simpler, use less energy

and incorporate many processor cores. Furthermore, Forth processor is implemented in VHDL language, which allows rapid prototyping in FPGA with simple instructions set to extend (30 times faster on FPGA Forth core than 68HC12 microcontroller at the same clock speed [**?**]).

- The NCU is connected to PEs through specific routers. We chose the specific router, Node elements (NE), to separate the management of communication and processing. First, this router is able to connect control unit to any PE following a well-defined parallel program. It defines the mask responsible for identifying PEs processing assistants. Based on this subnetting model, the enabled routers connect control unit to PEs involved in the processing. Second, this router is able to activate PE and control its running which offers the calculation flexibility and allows the decrease of the processing delays.

As shown in Figure 1, the subnetting model is fitted in mpp-SoC modified architecture, which has a single network control unit (NCU), and multiple nodes elements (NEs) combined with local processing element (PE), known collectively as Nodes. The NCU and NE array are connected through single level hierarchical bus and the PEs are connected together through an optional PE interconnection neighborhood network.

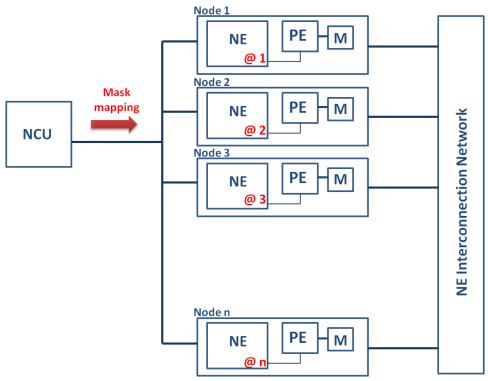

Fig. 1. Broadcast with mask model in massively parallel processing on chip architecture

2) Mask/Broadcast instructions: In addition to Forth processor instructions, we added mask/broadcast instructions to activate nodes responsible to process and broadcast parallel instructions from control unit to these specific node array. Table I sums up these instructions and their arguments.

Mask instructions

The major step in the broadcast with mask is the identification of active nodes. So at each node, we set up a mechanism for comparing the local address with mask plan. Consequently, if the result of the comparison is true, the active signal is set to 1 otherwise it is set to 0.

Once the active nodes are identified, the bit flag register BF inside each node becomes sensitive to mask instructions. The value of BF identifies if its own node participates in the processing or not. Consequently, only nodes with BF bit set to 1 will receive parallel processing instruction.

TABLE I
MASK & BROADCAST INSTRUCTIONS SET

Mask instruction
selbf: activate NEs in the selected set $BF \Leftarrow active$
selbfand: activate NEs in the intersection of sets $BF \Leftarrow BF\ and\ active$
selbfor: activate NEs in the union of sets $BF \Leftarrow BF\ or\ active$
selbfxor: activate NEs in the union of sets except the intersection part $BF \Leftarrow BF\ xor\ active$
Broadcast instruction
brdbf: broadcast parallels instructions to active NEs If BF = '1' then $R\_CMD \Leftarrow A(31...16)$; $R\_INE \Leftarrow A(15...0)$;
brdbfb: broadcast parallels instructions to inactive NEs If BF = '0' then $R\_CMD \Leftarrow A(31...16)$; $R\_INE \Leftarrow A(15...0)$;
brdall: broadcast parallels instructions to all the NEs $R\_CMD \Leftarrow A(31...16)$; $R\_INE \Leftarrow A(15...0)$;

NE: Node Elementary
BF: Bit Flag register
R_CMD: parallel instructions register
R_INE: NE instructions register

According to mask instructions the active nodes can process in one of four ways:

- Activate selected nodes (selbf). Depending on the signal active value, BF registers of selected nodes take on an enable bit.
- Activate intersection sets (selbfand). Nodes located in the intersection of previous set and the current set are activated.
- Activate union sets (selbfor). Nodes located in the union of previous set and the current set are activated.
- Activate "xor" sets (selbfxor). In same way as previously, nodes located in the union of previous set and the current set except the intersection part are activated.

Broadcast instructions

Once the mask is defined on the network, the active nodes have their bit flag set to 1 while the other nodes have their bit flag set to 0. Therefore, according to the broadcast instructions, the control unit will proceed as follows:

- Broadcast to selected nodes (brdbf). Only the nodes with BF register set to 1 will receive parallel instruction.
- Broadcast to inverse selected nodes (brdbfb). Only the deactivated node with BF register set to 0 will receive parallel instruction.

- Broadcast to all nodes (brdall). All nodes in our network will receive parallel instruction.

In each case of these instructions, the control unit can broadcast several parallel instructions to selected nodes while the mask is not changed. Once the broadcast is done to selected nodes, each active NE receives parallel instruction by 32bits input and stores the first sixteen most significant bits (MSB) in R_CMD register (contains control/command PE instruction) and the last sixteen least significant bits (LSB) in R_INE register (contains local memory address).

IV. Experimental results on subnetting model

In this section we describe one of subnetting models used in signal processing applications. Then, we present experimental results when executing this model with mppSoC modified architecture.

A. Red-Black broadcast description

Red-Black or checkerboard ordering method is used to resolve differential equations (Laplace equation with Dirichlet boundary conditions, Poisson-Boltzmann Equation,...) into massively parallel processing. In fact, this application is a good example for performance testing of SIMD architecture. This method is based on dividing grid nodes into red and black blocks then all blocks with same color are calculated in parallel after achieving the simultaneous and synchronous instruction broadcast.

This application was mapped to 256 Nodes (NEs + PEs), implemented in VHDL and targeted for a Xilinx Virtex6 (XC6VLX760) FPGA. The program below shows the code loaded in the NCU local memory. We use a combination of the added mask/broadcast instructions and the main Forth instructions among which "lit" instruction that allows retrieving the first element of the stack and pushing data stack.

Program of red-black broadcast

```
lit 0xAAAAAAAA   //find maskA(1010..)
selbf            //map maskA
lit 0x55555555   //find maskB(0101..)
selbfor          //map maskB
lit 0xD5F5021A   //find parallel
                 //instruction
brdbf            //broadcast parallel
                 //instruction
```

The description of this program is done in six steps detailed as below:

1) Set the first plan mask A. Match the mask A to Xmask and Ymask values.
2) Select the nodes whose addresses satisfy (X@ = Xmask and Y@= Ymask). Then set their bit flag to 1.
3) Set the second plan mask B. Again, match the mask B to the new Xmask and Ymask values.
4) Select the nodes whose addresses satisfy (X@ = Xmask and Y@= Ymask) in addition to the previously selected nodes. Then set their bit flag to 1.
5) Push parallel instructions code on stack.
6) Broadcast the parallel instructions to selected nodes.

The previous model description of red-black broadcast is rather easily generated. This section of control unit program shows that with a set of six instructions, we realize the subnetting of a NE grid (16x16 NEs) and the broadcast of parallel instructions to the active nodes. Because the Forth processor is 16 bits processor, reading each plan mask takes two clock cycles and the mapping process takes one clock cycle. Once the two combined, masks are tracing to our grid. The broadcast process takes one clock cycle to send parallel instruction from control unit to active nodes registers.

In order to implement the grid version of red-black broadcast, we have employed the subnetting technique which is graphically described in figure2.

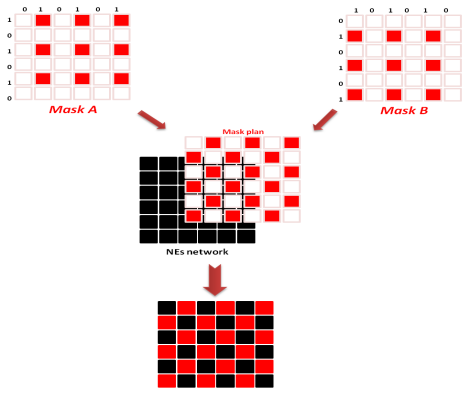

Fig. 2. Red-Black broadcast

B. Evaluation results

Using ISE tool [**?**], it was easy to prototype the broadcast with mask (or subnetting) model on the FPGA Virtex6 (XC6VLX760) device. Throughout the synthesis results presented in tableII and performance results presented in tableIII, we note that despite the slight rise of its occupied area, the subnetting approach provides a larger bandwidth (around 13% upper) than the one-to-all model. In fact, the broadcast with mask model shows that the use of masks will optimize the segmentation of our network and separate sensitive nodes from the rest of the network which better manage the NE grid and reduce data congestion.

The current densities of hardware logics no longer allow manual routing. Therefore, ISE tool integrates a specific tool for Place and Route (P&R) to map the logical schema intended by the designer and the material resources of the chip. Because the delay depends on the connections length between logic cells and the optimization algorithms of the P&R are not deterministic, obtained performance (maximal frequency and power supply) are variable depending on the design. Therefore, Xpower tool gives a view of optimized power consumption of the system. Thus, tableIII shows that with subnetting model, we save around 35% on energy supply and 5 times on broadcast latency compared to the simple one-to-all model. In

TABLE II
SYNTHESIS RESULTS

	Boradcast with mask (subnetting) model	One-to-all model	FPGA resources
Register	$41090_{4\%}$	$41056_{4\%}$	948480
LUT	$257747_{54\%}$	$242768_{51\%}$	474240
LUT-FF	$26897_{11\%}$	$28853_{11\%}$	268203
BUFG/BUFGCTRL	$3_{9\%}$	$2_{6\%}$	32
Memory	$10256_{7\%}$	$10256_{7\%}$	132480

LUT: Look-Up Table
LUT-FF: LUT-FIFO couple
BUFG/BUFGCTRL: Buffers

TABLE III
PERFORMANCE RESULTS

	subnetting model	One-to-all model
Fmax (Mhz)	205.999	182.846
Bandwidth (MB/s)	786	697
Power supply (W)	3.214	4.447
Latency (Cycles)	3	16

fact, the traditional broadcasting in mppSoC architecture with a grid of 16 x 16 Nodes needs specific instructions to realize activation and deactivation of 256 elements alternating. This is very tedious and consumes a lot of energy and time (16 clock cycles), while using broadcast with mask allows the subnetting of our network into active/inactive nodes in two clock cycles and the broadcasting in one time (3 clock cycles).

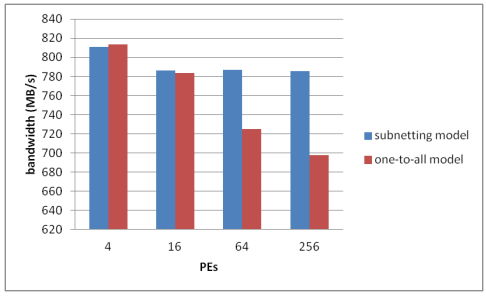

Fig. 3. Influence of broadcast models on bandwidth

These models were implemented with different network levels. As shown in figure3, the broadcast with mask model gives a large bandwidth than the broadcast with one-to-all model. The gap between the bandwidth rates presented in figure3 is more significant with the increasing of the number of PEs. In fact, the subnetting model mapped on mppSoC with 256 PEs gives a large bandwidth than the other one. It is therefore an interesting technique to increase the architecture scalability and to avoid broadcast bottleneck.

However, the broadcast in small architecture doesn't need the subnetting step. In this case, traditional technique seems to be more efficient in term of area cost and bandwidth. Thus, according to the application, the designer has to make the right choice between architecture and the model of broadcast in order to optimize the whole system's performances.

V. CONCLUSION

This paper introduces a new paradigm of broadcast technique with mask. It consists in subnetting the network of elementary processing nodes which is responsible of controlling the synchronous communication through regular neighborhood interconnection network and managing the processing through elementary processors. The set of broadcast instructions are also defined to assure the communication between Control Unit and nodes-grid. The proposed model is entirely described in VHDL language, in order to allow accurate description and fast verification, before synthesis tools translate the design into real hardware (FPGA). Our model has been validated on mppSoC architecture with red-black ordering method. This work represents a significant step in the techniques of broadcasting on a massively parallel processing architecture.

Our next work is to generalize this technique of broadcast in commonly used massively parallel systems and to propose a new execution model, which supports SIMD and MIMD execution in the same time. This parametric and generic model, called "Synchronous Communication Asynchronous Computation" (SCAC) will be prototyped to satisfy the PPA constraints (Performance, Power, Area) imposed by the market of embedded applications and to help the designer to perform an execution model for a specific architecture for a given application.