



Alexander Logvinenko (logvynen@uni-wuppertal.de), Carsten Gremzow (gremzow@uni-wuppertal.de), Dietmar Tutsch (tutsch@uni-wuppertal.de)

Introduction

Increasingly, multicore processors are used as a means to achieve increased computational power in the modern processing chip. Multicore chip effectiveness is, however, hampered by difficulties enabling efficient interaction between processor cores that have to communicate with each other and other IP-Cores e.g., cache, memory, I/O controllers, audio/video interfaces, and communication interfaces. In the context of communication, the processors and other modules on the same chip, are called network nodes. Designing a high-performance network on chip (NoC) that enables an efficient interaction between these nodes represents a serious challenge in the domain of multicore chip manufacturing. The key factor in how effectively NoCs will operate is how they alter the network topology to optimize the traffic distribution to destination nodes, creating an appropriate network architecture that improves the performance of the entire system. Current systems utilize fixed architecture, which is adversely affected by changes in traffic flow and can thus distribute traffic inefficiently. Network on chip architectures are often based on a general-purpose design, able to support a wide range of applications with varied communication and computation profiles. However, applications with irregular traffic profiles will have their performance improved when specialized network topologies are employed. This poster proposes a new reconfiguration architecture: RecMIN (Reconfigurable Multi- Interconnection network) that would allow the NoCs to individually reconfigure according to the needs of traffic.

Generator	$P_{tr}(G_i)$	$P_{rec}(T_{10})$	$P_{rec}(T_{11})$	$P_{rec}(T_{12})$	$P_{rec}(T_{rst})$
G_0, G_1	0.65	0.65/16	0.65/16	0.65/16	0.65/16
G_2, G_3	0.65	0.65/16	0.65/16	0.65/16	0.65/16
$G_4 - G_7$	0.65	0.65/16	0.65/16	0.65/16	0.65/16
$G_8 - G_{15}$	0.65	0.65/16	0.65/16	0.65/16	0.65/16

Tabelle I: Load 1 (0 - 5 000 and 15 000 - 30 000 clocks)

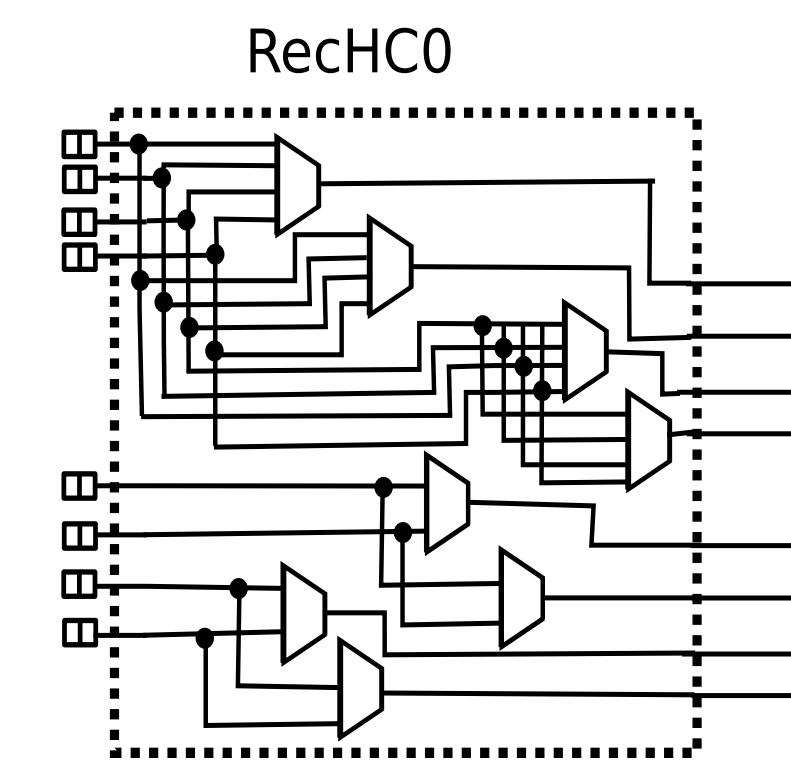


Fig. 1: RecHC0

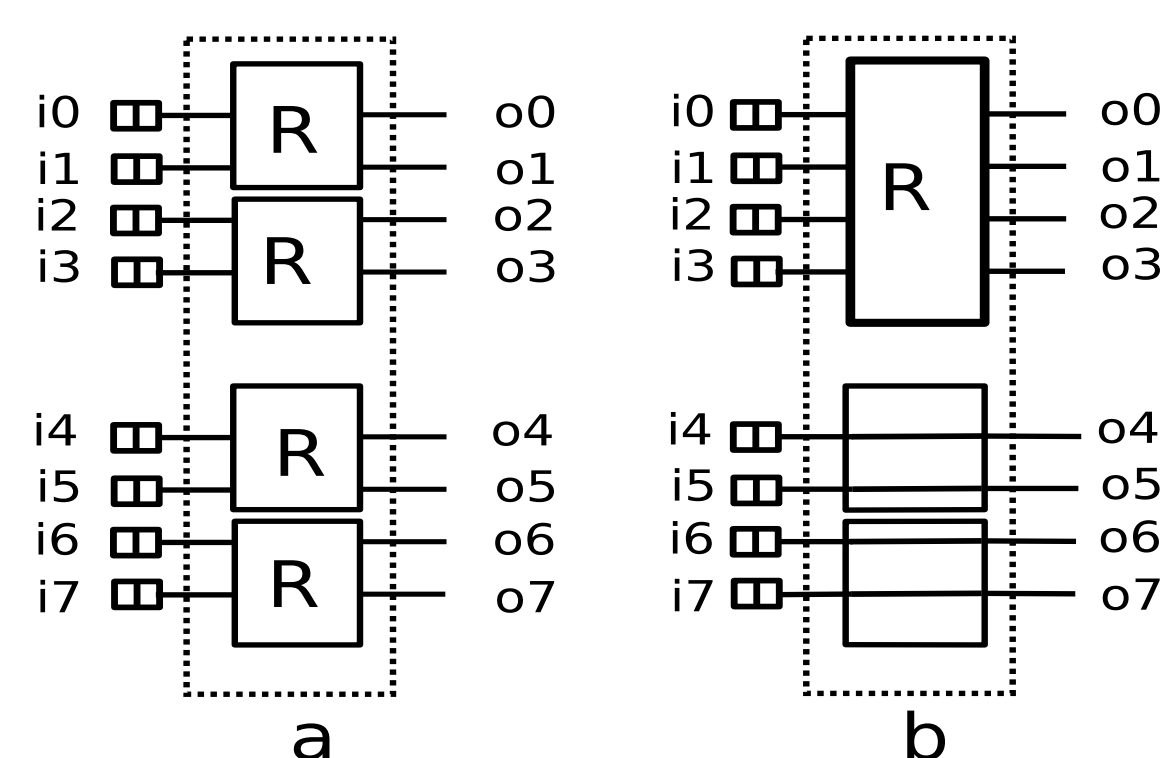


Fig. 2: RecHC in two modes.

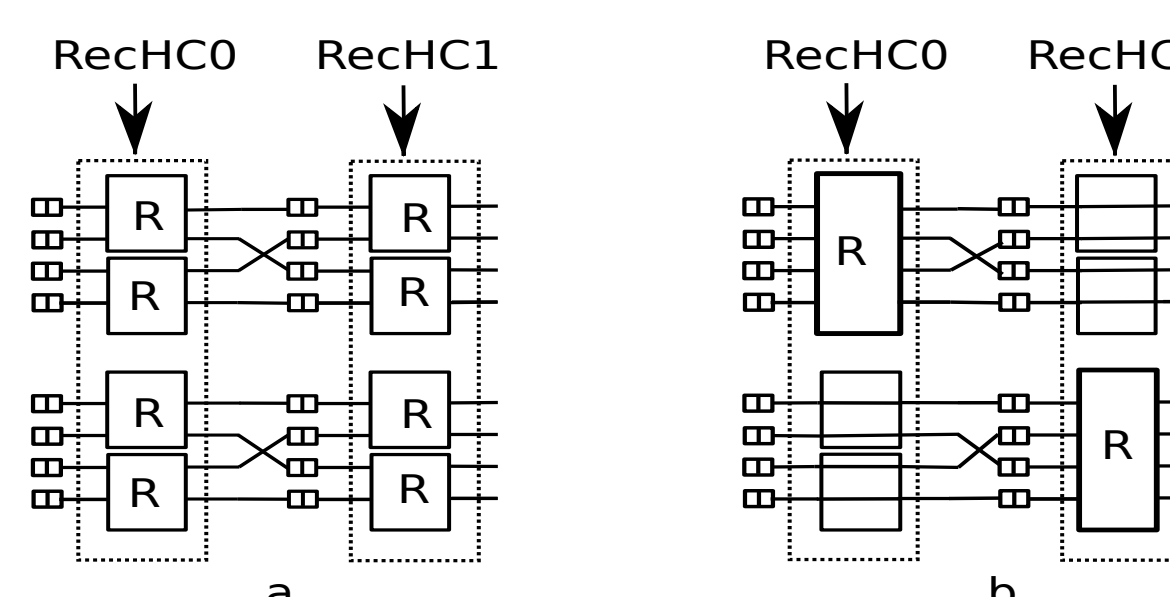


Fig. 3: RecCell a: unfolded mode, b: folded mode

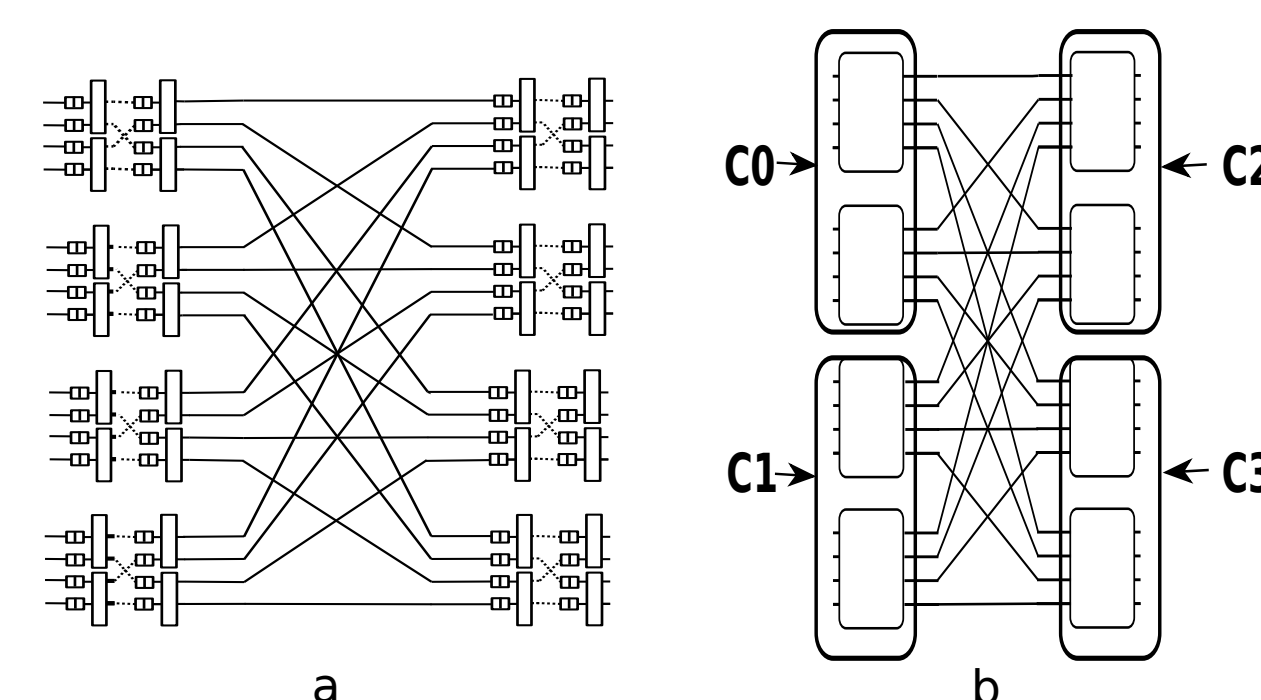


Fig. 4: RecMIN with 16 inputs/outputs

RecCell

The main problem of NoC as compared to the full connection of all inputs/outputs is the chance of bottlenecks to arise given certain traffic structures. In this work MIN architecture is used, which is built out of 2x2 routers. One of the characteristics of MIN is that all the traffic loads have to pass through all the stages of the MIN. Especially for asymmetrical traffic, the connection wire between the stages can lead to tailbacks.

However, this poster suggests a solution for this problem which would solve the problem of bottlenecks in two out of three possible cases. The proposal is to create the MIN not from the 2x2 routers as usual, but from specific reconfiguration half cells - RecHC (Reconfiguration Half Cell). The architecture of this cell is given in the Fig. 1.

RecHC (Fig. 2) has 8 inputs and 8 outputs. In front of each input one buffer element is located. Each half-cell can be used in one of these two possible modes: In the first mode (Mode A) Fig. 2a the RecHC consists of four independent 2x2 routers. In the second mode (Mode B) (Fig. 2b) there is however just one 4x4 router in the upper part of the cell, and four simple wire connections without any logic in its bottom part. If a RecHC changes the mode from A to B (Fig. 2), then packets which arrive in the upper part of the Half Cell are distributed correctly without problems. Though, in the bottom part of the RecHC problems may arise, since in the mode B no redirection takes place, and the packets are transferred straight forward (Fig. 2). I. e., after switching from mode A to mode B some packets in buffer of input i4 that are addressed to output o5 have no possibility to arrive at their targets (e. g. IP cores). Therefore usage of two half cells simultaneously is to be preferred.

The two RecHCs are put together (the second one upside down), to form one reconfiguration cell - RecCell (Fig. 3). If both of RecHCs that build RecCell are put into the Mode A, then the construction leads to two independent MINs (Fig. 3a), with 4x4 inputs-outputs and 2x2 routers each. If the two RecHCs are put in mode B, then two independent 4x4 routers emerge (Fig. 3b). The other two combinations (AB and BA) are meaningless and therefore are not used. So a full cell has two possible reconfigurations: folded (BB) and unfolded (AA).



Generator	$P_{tr}(G_i)$	$P_{rec}(T_{10})$	$P_{rec}(T_{11})$	$P_{rec}(T_{12})$	$P_{rec}(T_{rst})$
G_0, G_1	0.4875	0.2/16	0.3	0.2/16	0.2/16
G_2, G_3	0.3875	0.2/16	0.2/16	0.2	0.2/16
$G_4 - G_7$	0.55	0.1	0.2/16	0.2/16	0.2/16
$G_8 - G_{15}$	0.2	0.2/16	0.2/16	0.2/16	0.2/16

Tabelle II: Load 2 (5 000 - 15 000 clocks)

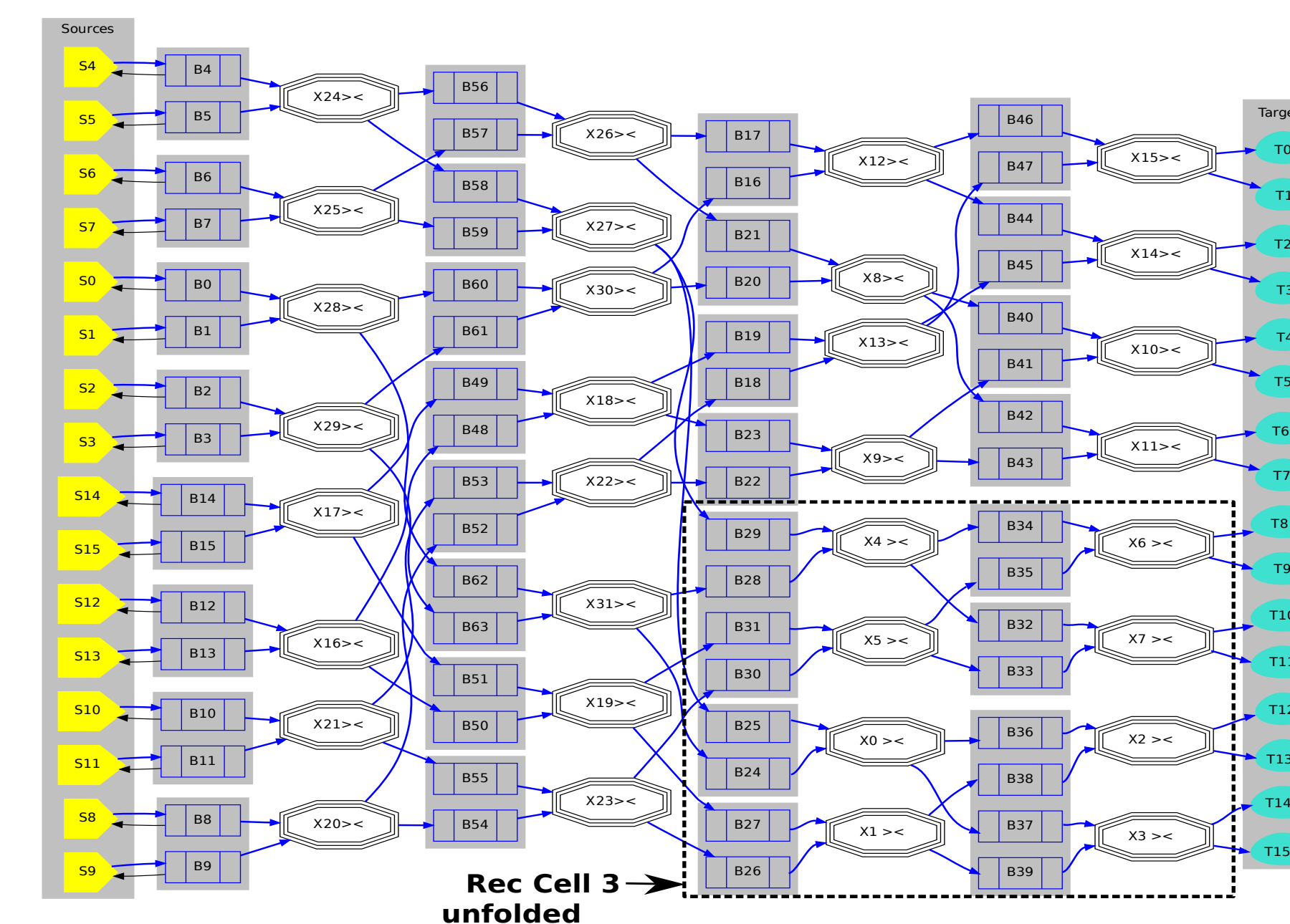


Fig. 5: RecMIN with 16 inputs/outputs. Output of simulator RecSim

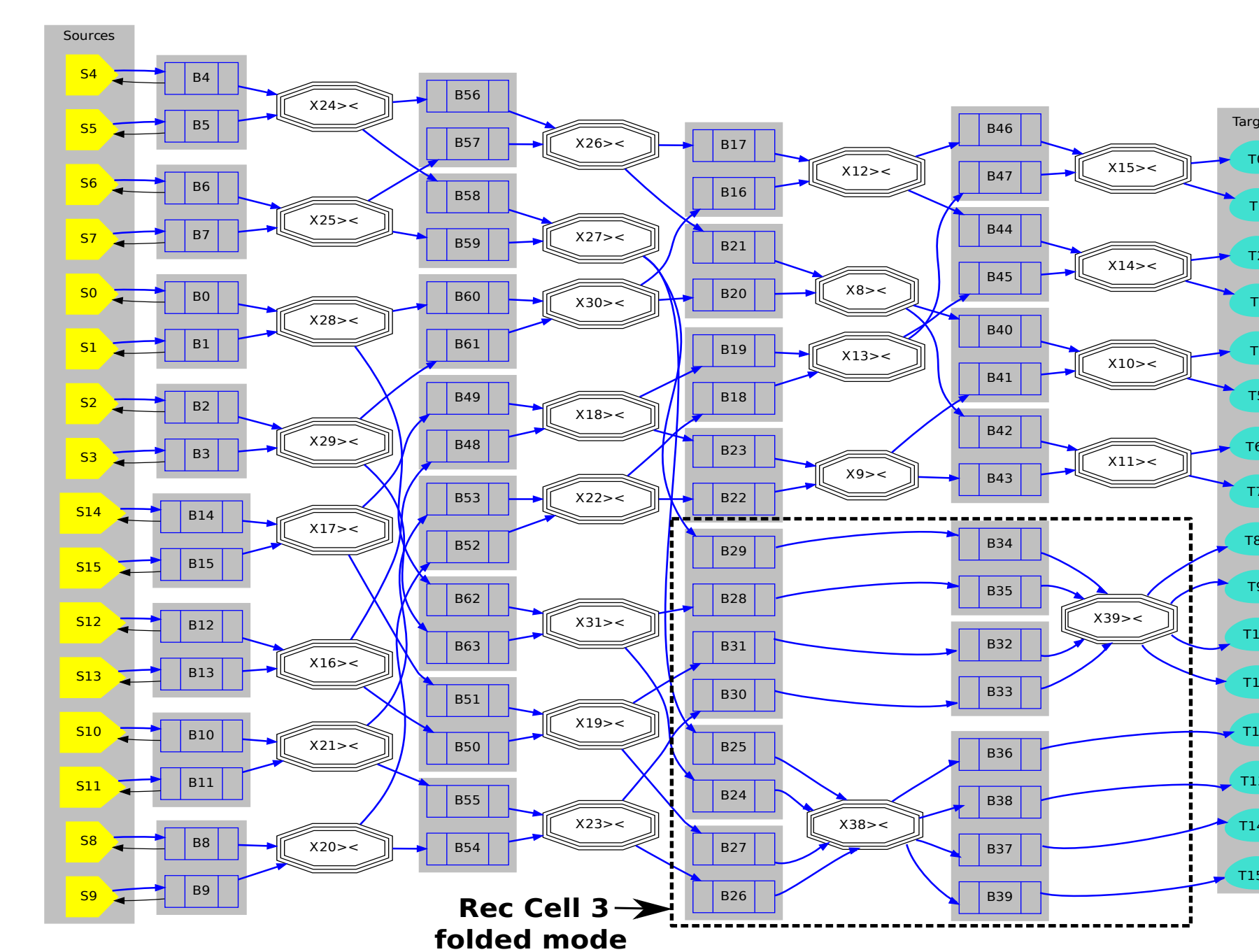


Fig. 6: RecMIN with 16 inputs/outputs. Output of simulator RecSim

Reconfiguration

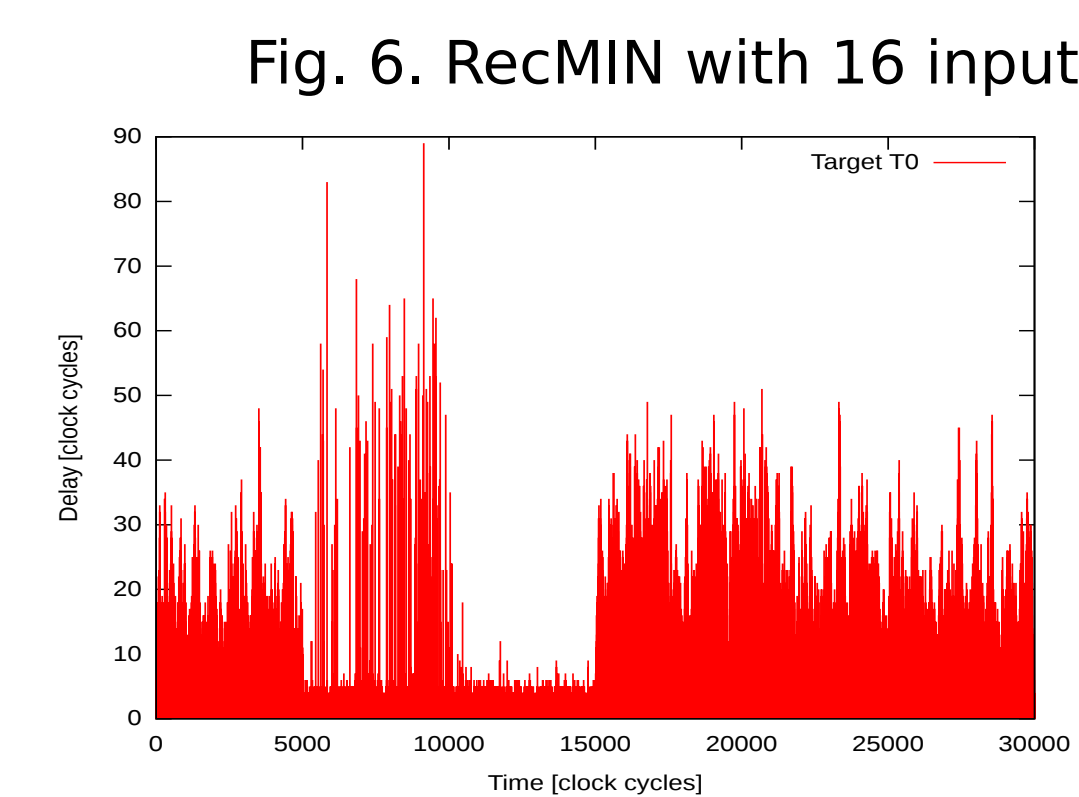


Fig. 7: Measured packet delay in Target 0

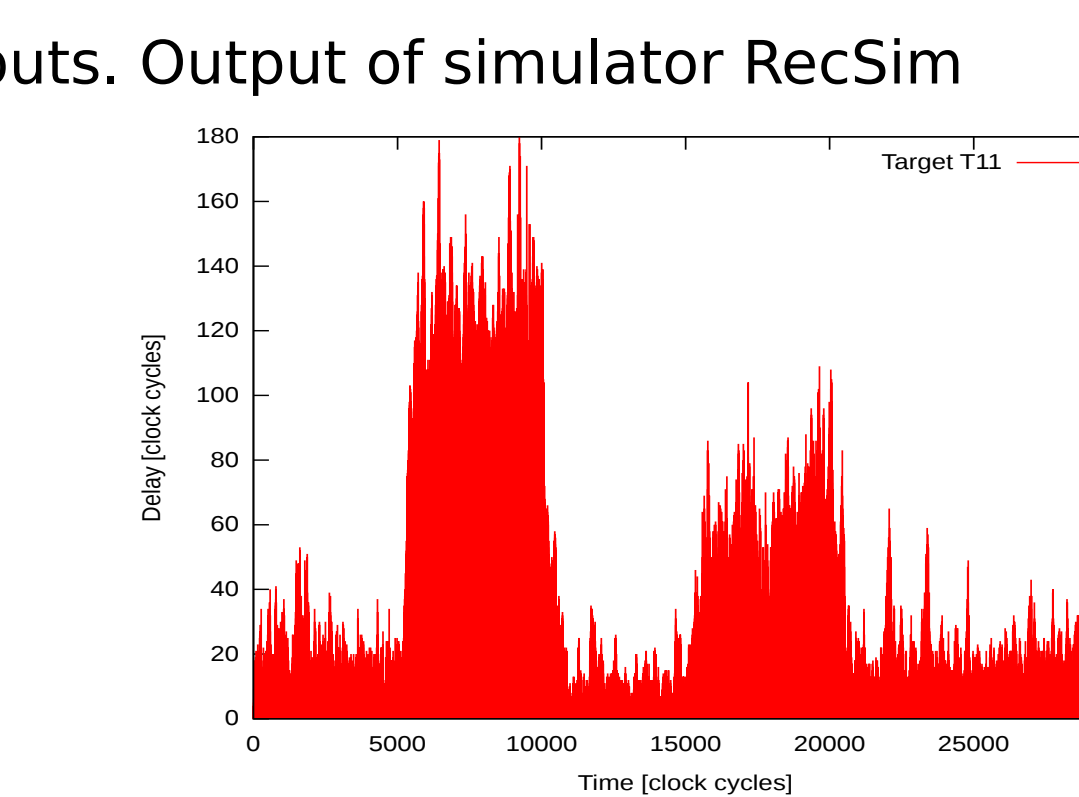


Fig. 8: Measured packet delay in Target 11

RecMIN architecture

With RecCells, it is possible to build a MIN. The resulting structure is called RecMIN. If the number of 2x2 switches in MIN is divisible by 16, the entire network can be built from reconfiguration cells. Otherwise it is necessary to use two non-reconfigurable 2x2 switches in order to connect the reconfiguration cells. Therefore RecMIN with an arbitrary number of 2x2 routers can be implemented.

In this poster the RecMIN with 16 inputs outputs is used as an example for RecMIN architecture (Fig. 4). This RecMIN can be build out of four RecCells: C0,C1,C2 and C3.

It can be said that if the traffic unfortunately generates a bottleneck in one of non-reconfigurable wires (Fig. 4, the non-dotted line marked router connection), the reconfiguration will not help. But usually the designer of the NoC knows, the application for which the network is to be designed, and so can pre-arrange the most expected bottleneck-wires inside the reconfiguration cells.

The other way around, the 4x4 router would have less throughput than a 2x2 router. So for the symmetrical high load (more than 0.63 flits per clock cycle), the 4x4 routers will automatically become NoC bottlenecks. In this case a back reconfiguration of the RecCells to unfolded mode is necessary.

Results

We simulated the RecMIN with 16 inputs/outputs (Fig. 4) using RecSim (simulator for reconfigurable network on chip topologies) and tested its efficiency with two different loads. (Simulation parameters: buffer size of 16 phits for each buffer, conflict resolution algorithm for each router is "random choice", each packet consists of one flit, a flit equals the size of a phit). This demonstrated the potential of reconfiguration architecture RecMIN to improve the performance of NoC. We alternated loads through the network periodically; Load 1 (Table I) and Load 2 (Table II) as samples of changing traffic profiles.

Fig. 7 and Fig. 8 show the effects of reconfiguration, as measured packet delay in Output 0 and Output 11 of our NoC. During the first 5000 time cycles, Load 1 was being sent through the network. After this point, the traffic switched to Load 2. This immediately led to an overload in the network, resulting in a much higher packet delay. A further 5000 time cycles in this state are recorded, which gave time to demonstrate that the overload results in a continuous delay, until the network was reconfigured. Having demonstrated this, the network was reconfigured and the packet delay reduced. Continuing the experiment, after 5000 time cycles of the efficient, reconfigured network processing of Load 2, it reverted to Load 1 but again, the reconfiguration of the network was delayed, demonstrating that once again, the packet delay rose considerably. After 5000 time cycle, we reconfigured the network, and once again, the packet delay fell commensurately.

Fig. 5 and Fig. 6 show the RecMIN topology as a graphical output of the RecSim engine. The simulation shows that in order to avoid the bottlenecks for Load 1, the RecCell3 has to operate in unfolded mode (because the 2x2 router has better throughput than 4x4 router), and for the Load 2 the RecCell3 has to operate in folded mode (otherwise traffic generates a bottleneck in the input of buffer B32). The other RecCells of RecMIN do not need any reconfiguration during the simulation.