Optimized Design of 2D Mesh NOC Router using Custom SRAM & Common Buffer Utilization

Bhavana Pote[1],V. N. Nitnaware[2],and Dr. S. S. Limaye[3]

[1] Department of Electronics, Ramdeobaba Kamla Nehru College of Engg, Nagpur, India
bhavanaspote@gmail.com

[2] Department of EDT, Ramdeobaba Kamla Nehru College of Engg, Nagpur, India
nitnawarevn@rknec.edu

[3] Jhulelal Institute of Technology, Nagpur, India
ss_limaye@gmail.com

Abstract:

With the shrinking technology, reduced scale and power-hungry chip IO leads to System on Chip. The design of SOC using traditional standard bus scheme encounters with issues like non-uniform delay and routing problems. Crossbars could scale better when compared to buses but tend to become huge with increasing number of nodes. NOC has become the design paradigm for SOC design for its highly regularized interconnect structure, good scalability and linear design effort. The main components of an NoC topology are the network adapters, routing nodes, and network interconnect links. This paper mainly deals with the implementation of full custom SRAM based arrays over D FF based register arrays in the design of input module of routing node in 2D mesh NOC topology. The custom SRAM blocks replace D FF(D flip flop) memory implementations to optimize area and power of the input block. Full custom design of SRAMs has been carried out by MILKYWAY, while physical implementation of the input module with SRAMs has been carried out by IC Compiler of SYNOPSYS.The improved design occupies approximately 30% of the area of the original design. This is in conformity to the ratio of the area of an SRAM cell to the area of a D flip flop, which is approximately 6:28.The power consumption is almost halved to 1.5 mW. Maximum operating frequency is improved from 50 MHz to 200 MHz. It is intended to study and quantify the behavior of the single packet array design in relation to the multiple packet array design. Intuitively, a common packet buffer would result in better utilization of available buffer space. This in turn would translate into lower delays in transmission. A MATLAB model is used to show quantitatively how performance is improved in a common packet array design.

Keywords *: 2D mesh, virtual output queuing, HOL blocking, FIFO, DDC file, GDS format.*

1. Introduction:

A packet-switched 2-D mesh is the most used and studied topology so far. It is also a sort of an average NOC currently. Good results and interesting proposals provoke design engineers to use this topology as the base**[1]** .The key research problems in the design of NOCs include but are not limited to topology, channel width, buffer size, floorplan, routing, switching, scheduling, and IP mapping**[2].** Additionally, **[3]**lists research issues to be application modeling and optimization, NOC communication architecture analysis and optimization, NOC communication architecture evaluation, and NOC design validation and synthesis.

The most important metrics for NOCs are application runtime, silicon area, power consumption, and latency. All these are to be minimized and usually appropriate trade-off is sought**[4].** The required silicon area is the most commonly reported value (77%) followed by latency (55%) and maximum operating frequency (50%). The other metrics have lower occurrence**[1].**

In this regard, the current work is related to optimization of buffers in the router design so as to achieve lower silicon area, lower power and higher operating frequency.

The input block in the design of router consists of six major components: the packet array, the linked list array, the destination head array, the destination tail array, the free-list FIFO, and a shift register, see fig. 3.1. Four of these six components are conventional memory elements. In a standard cell based design, memory elements are realized using D flip flops in the standard SYNOPSYS Library. If we consider a NAND gate implementation of a D flip flop with no RESET or SET inputs, we require 28 MOS transistors to realize one D flip flop, see fig. 1.1. A more area efficient implementation of memory is through the use of SRAM cells. Each SRAM cell is implemented using 6 transistors, see fig. 1.2. Therefore, memory realization using SRAM is more efficient compared to D flip flops. However, standard cell based approach to ASIC design does provide SRAM standard cells because of the many possible configurations of width and depth. SRAM design is carried out using full custom approach to ASIC design. By combining standard cell based and full custom ASIC design, D flip flops can be replaced by SRAM, improving the area efficiency of the input block (Fewer transistors - less area required for equal amount of memory).

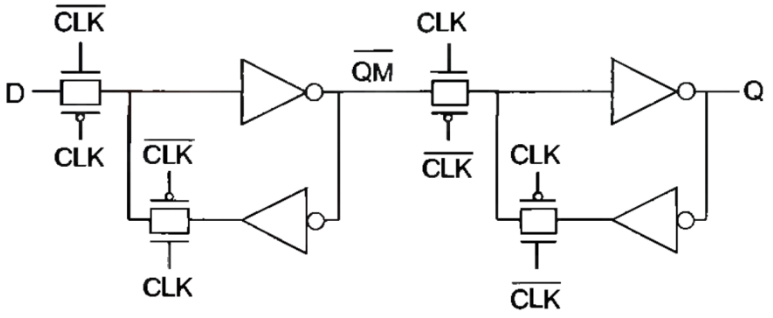


Fig. 1.1 DFF schematic

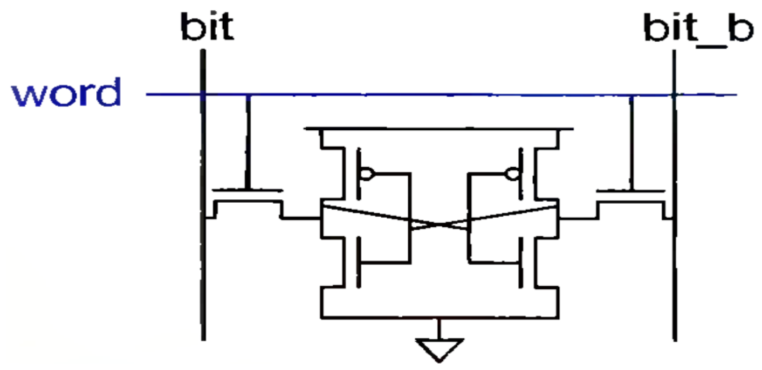


Fig. 1.2 SRAM schematic

The queues are maintained in a packet array. Queue size is dynamically determined depending on the arrival pattern of the data. If more data is destined for output port "m", then correspondingly, more buffer space, and hence, a longer queue is maintained for data packets to be routed to output port "m," subject to the maximum space available in the packet array (Fig. 1.3).

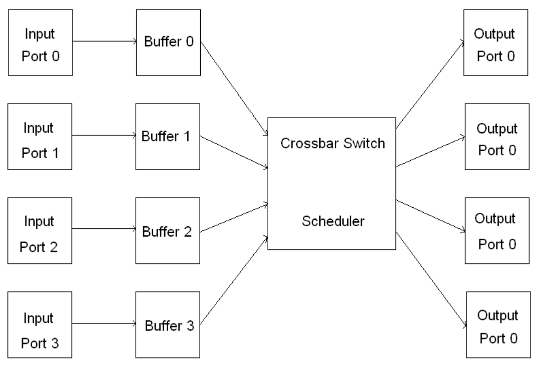


Fig. 1.3 Each input port has its own buffer

An alternative design is based on using a common packet for all the input ports. For example, if the crossbar switch consists of four input ports, then the original design calls for four packet arrays. The proposed design would utilize one common packet array for all the four input ports (Fig.1.4)

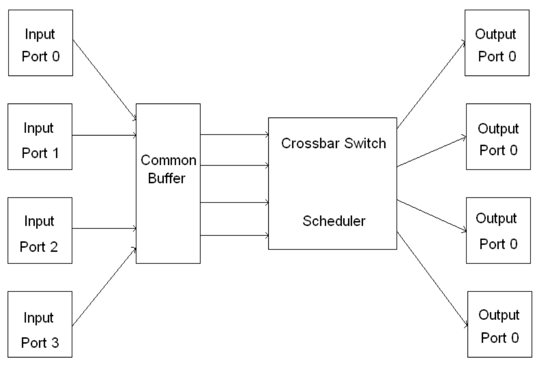


Fig. 1.4 Common buffer shared between all input ports

It is intended to study and quantify the behavior of the single packet array design in relation to the multiple packet array design. Intuitively, a common packet buffer would result in better utilization of available buffer space. This in turn would translate into lower delays in transmission.

2. Introduction to A 2D Mesh NOC:

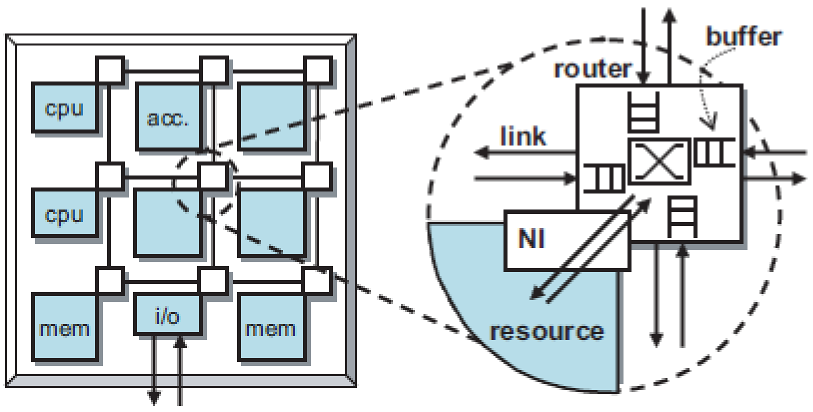


Fig. 2.1 : A 2D Mesh NOC structure

A NOC consists of routers/switch, links, and network interfaces (Fig. 2.1). Routers direct data over several links (hops). Routers further consists of a scheduler, buffer to store the incoming data packet and the crossbar. Topology defines their logical lay-out (connections) whereas floor plan defines the physical layout. The function of a network interface (adapter) is to decouple computation (the resources) from communication (the network). Routing decides the path taken from source to the destination whereas switching and flow control policies define the timing of transfers[**2**][**5**]. Task scheduling refers to the order in which the application tasks are executed and task mapping defines which processing element (PE) executes certain task. IP mapping, on the other hand, defines how PEs and other resources are connected to the NoC.

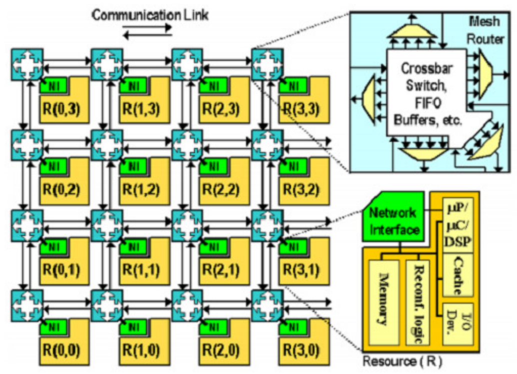
For illustrative purposes, Fig. 2.1 shows an example SOC with a NOC and nine heterogeneous IP blocks that are CPUs, memories, input/output devices, and HW accelerators.

2.1 Router Structure

NOC architectures are based on packet-switched networks, see figure 2.2a. This has led to new and efficient principles for design of routers for NOC[**6**]. Assume that a router for the mesh topology has four inputs and four outputs from/to other routers, and another input and output from/to the Network Interface (NI). Routers can implement various functionalities - from simple switching to intelligent routing. Since embedded systems are constrained in area and power consumption, but still need high data rates, routers must be designed with hardware usage in mind. For circuit-switched networks, routers may be designed with no queuing (buffering). *For packet-switched networks, some amount of buffering is needed, to support bursty data transfers.*

Such data originate in multimedia applications such as video streaming. *Buffers can be provided at the input, at the output, or at both input and output***[7].**

Various designs and implementations of router architectures based on different routing strategies have been proposed in the literature. Wolkotte et al. proposed a circuit switched router architecture for NOC**[8],** while Dally and Towles proposed a packet switched router architecture **[9].** Albenes and Frederico provided a wormhole-based packet forwarding design for a NOC switch**[10].**



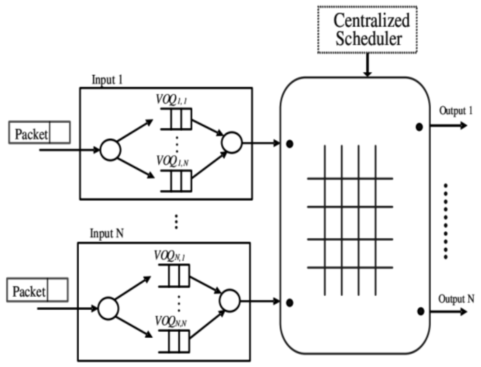


Fig 2.2 (a) NOC Router **(b) Router Components**

In this paper, the buffers in the design of the routers are based on the principle of virtual output queuing since it is simple and reduces the risk of Head of Line Blocking**[11] [12] [13].**

In this paper, the scheduling policy embodied in the router is based on Iterative SLIP algorithm. iSLIP uses round-robin to choose on port among those contending. This permits simpler hardware implementations compared, besides making iSLIP faster. iSLIP achieves close to maximal matches after just one or two iterations.. iSLIP achieves 100% throughput under uniform traffic and the round robin policy ensures fairness among contenders.

Even though its behavior may be unstable under bursty traffic, iSLIP is commonly implemented in commercial switches due to its simplicity**[14].** This algorithm becomes more silicon area efficient if it is implemented with its folding concept**[15].**

3. Implementation of input module with D FF based array vs CUSTOM SRAM.

The traditional design of input module consists of D FFs based register arrays**[16],** see fig. 3.1 The same function of input module we intended to achieve using full custom SRAM replacing the register arrays. The input module with arrays based on D FFs is now replaced by the input module with full custom SRAMs, see fig.3.2

The input block is synthesized using the Synopsys 90 nm EDK standard cell library. Memory elements are synthesized using D flip flops using the standard cell library. In order to save silicon area, D flip flop memory is replaced by custom built SRAM. The SRAM provided by Synopsys in the 90 nm process are available in sizes (width x depth) of 8x16, 8x32, and 128x64.

However, the input block requires packet array memory of size 72x32. This is realized using SRAM of size 128x64. The free-list FIFO requires a 6x32 memory. This is implemented using an SRAM of 8x32. The destination head and destination tail arrays are of size 6x8. These are replaced by SRAM of size 8x16. The SRAM used is underutilized. However, due to more compact implementation, overall design area is greatly reduced, as shown by results in the following section.

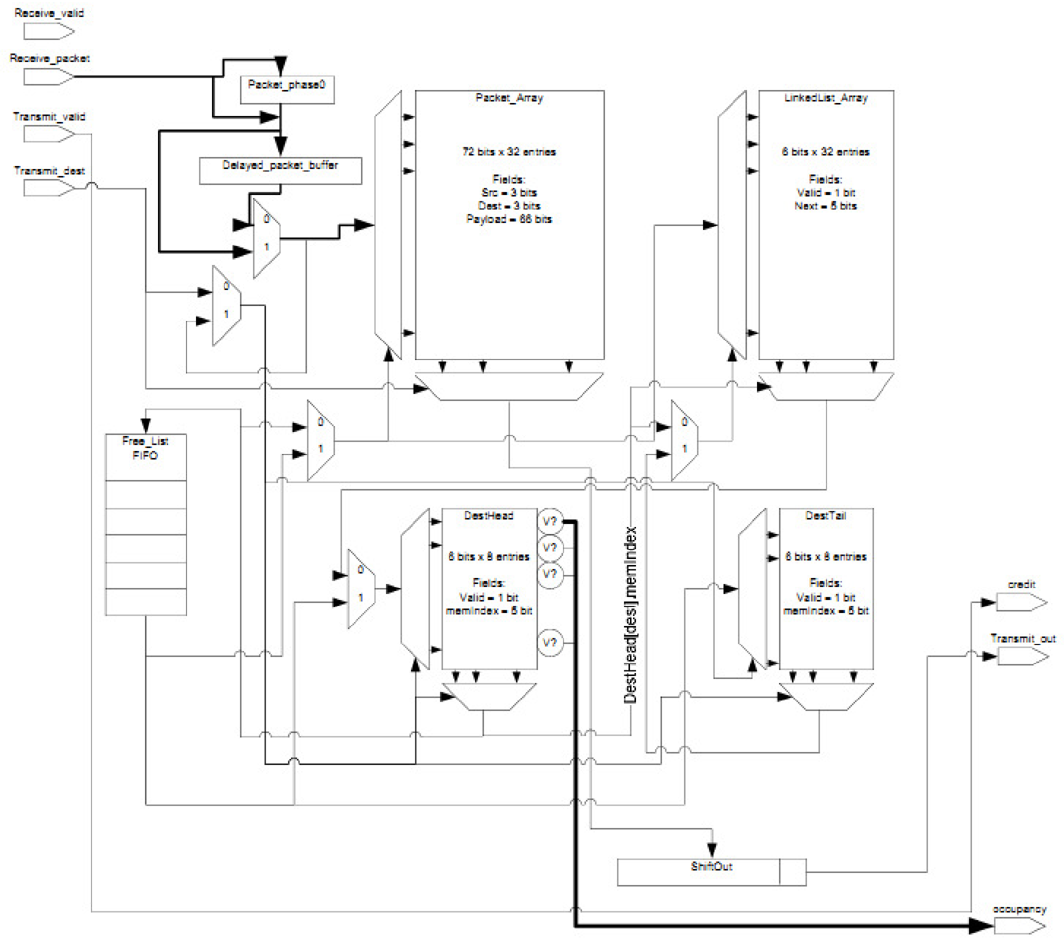


Fig.3.1 Input module with D FF based register arrays **[17]**

Custom SRAM IP is generated by full custom mask layout. Synopsys provides SRAM macros in LEF format. In order to integrate SRAM into the input block, this LEF file is used to generate CEL, FRAM, and LM views. Synopsys Milkyway is used to process the LEF file to create CEL, FRAM, and LM views. These views combine to produce a reference library for IC Compiler.

The layout of the input block using SRAM requires CEL, FRAM, and LM views of the 90nm standard library as well as the SRAM macro. The design file containing the input block gate level netlist is a DDC file. This file is generated by Design Compiler from RTL code by applying appropriate constraints.

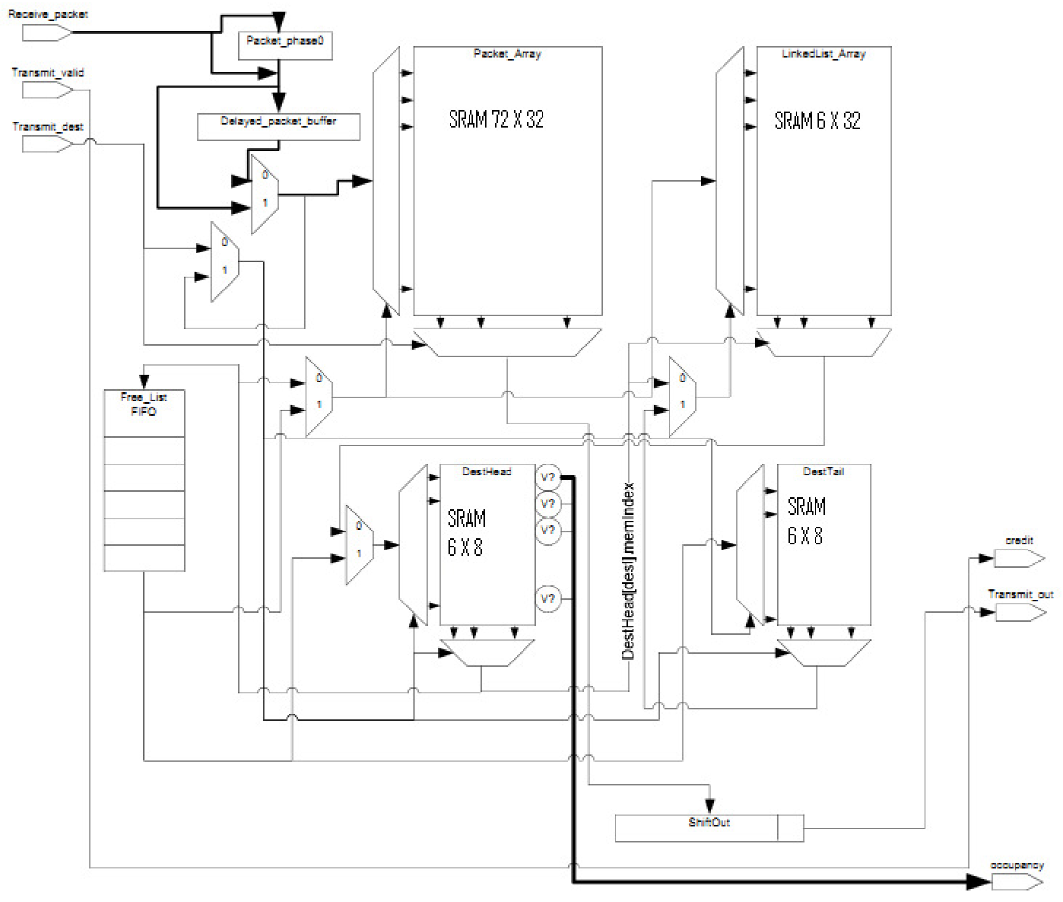


Fig.3.2 Input module with SRAM based arrays

The DC constraints applied to Design Compiler for the modified input block are:

Clock = 5 ns
Input Delay = 2 ns
Output Delay = 2 ns

The input and output delays are derived from the clock using a time budget of 40%.

The layout is generated from the gate level netlist and the reference libraries. Additionally, the TLU+ capacitance models are required by IC Compiler.

3.1 Steps on IC Compiler [18]

The following steps are executed on IC Compiler to derive the layout:

1. A new Milkyway library database is created for the input block.

2. Initial floorplan is created where the start row and row orientation is specified. Additionally, the spacing between the IC core and input/output pad ring is specified.
3. Power grid is initialized for VDD and VSS.
4. Rectangular rings are created for VDD and VSS. For each segment, the metal layer, segment width, and segment offset are specified.
5. Floorplan placement is executed. All cells and IP cores are placed within the IC core boundary.
6. Power straps are generated for VDD and VSS. The direction, width, and metal of power straps is specified.
7. Standard cells are prerouted for horizontal connection.
8. Clock tree is synthesized.
9. Detailed routing is performed.

The layout is ready to be streamed out in GDSII format.

3.2 Single Buffer of size 128 packets versus 4 buffers of size 32 packets:

The block diagram of the Simulink model is given below.

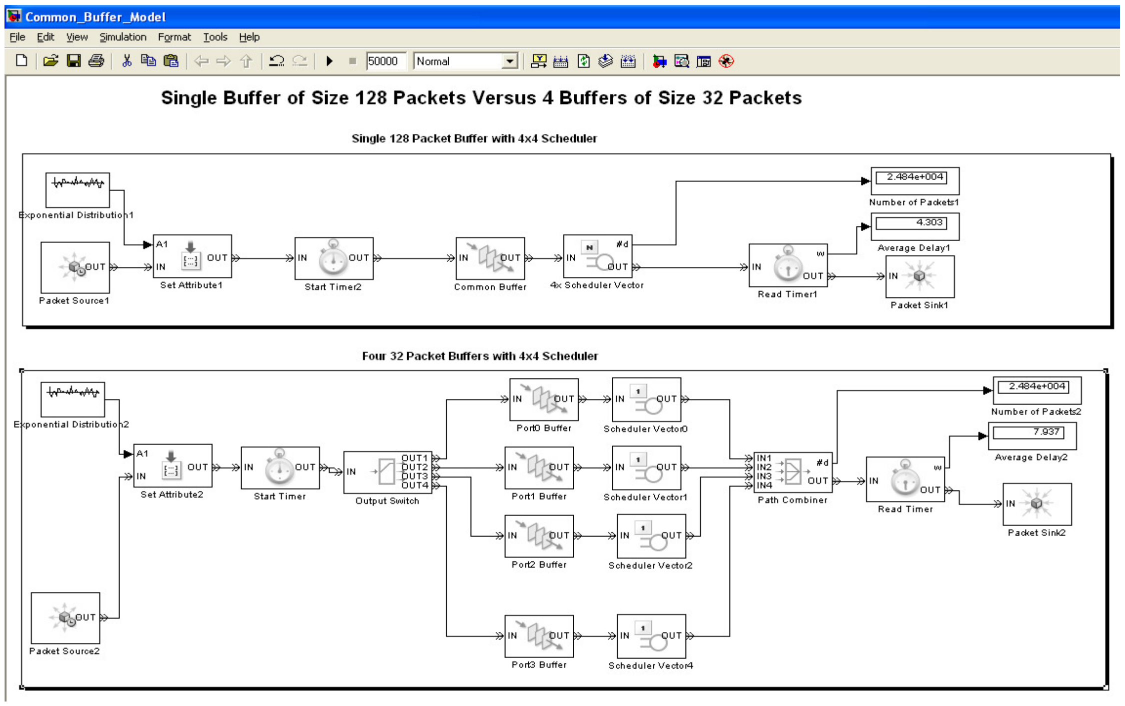


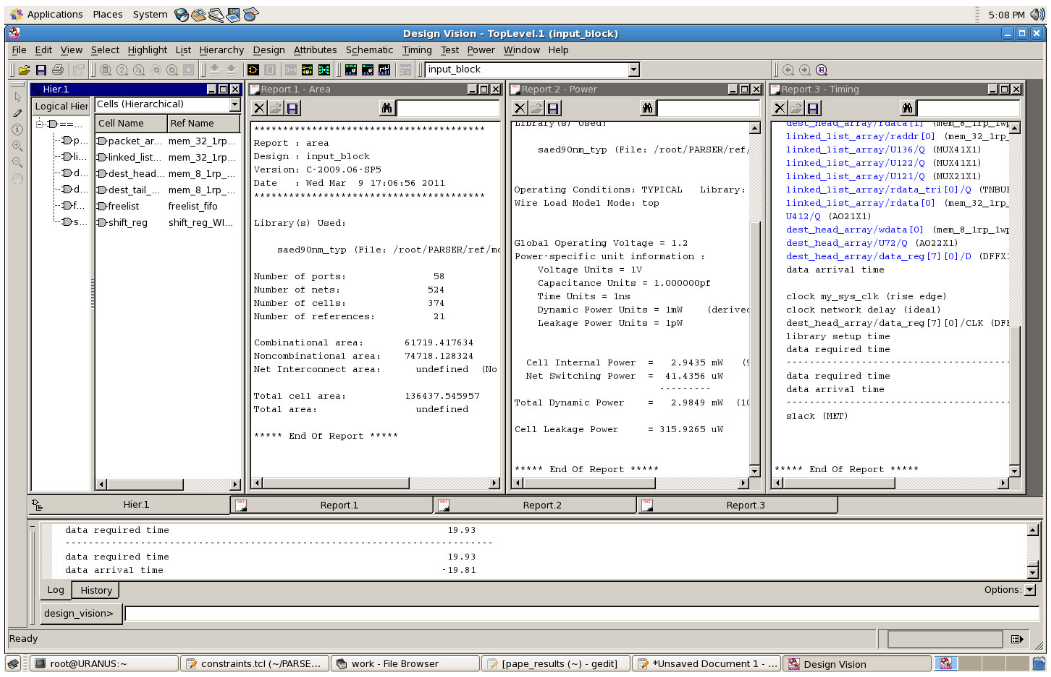
Fig. 3.3 Simulink model for single buffer vs 4-buffers.

The first block is labeled “Exponential Distribution1.” This block specifies packet arrival time. The packet arrival pattern is an exponential distribution. The block labeled “Packet Source1” generates packet events. The “Set Attribute1” block combines the effects of “Exponential Distribution1” with “Packet Source1” to generate packet entities at time intervals specified by the exponential distribution mentioned above. The arrival of a packet event at the “Start Timer2” block causes the simulation timer to start. Generated packets are stored in a common 128 packet buffer designated “Common Buffer.” The packets leave the common buffer when they are serviced by the scheduler vector. The scheduler vector is generated for four input ports by “4x Scheduler Vector.” The total number of packets served are recorded by the “Number Served1” block. Whenever a packet leaves the buffer, the departure time is recorded by the “Read Timer.” The packet exits the simulation flow through the “Entity Sink1” block. The average time spent by the packet in the buffer is captured by the “Average Delay1” block.

The behavior of the dedicated 32 packet buffer model differs only in two components, “Output Switch” and “Path Combiner.” The “Output Switch” block demultiplexes the generated packets into their respective input port packet buffers. The “Path Combiner” aggregates the output stream to help calculate total number of packets served and average time spent by packet waiting for service.

The simulation was run for 50000 packets. The packet generation rates for both models are identical, using the exponential distribution for inter-arrival times. Multiple simulation runs were performed to verify the average delays observed.

4. RESULTS:


Applications Places System
5:08 PM
Design Vision - TopLevel.1 (input_block)
File Edit View Select Highlight List Hierarchy Design Attributes Schematic Timing Test Power Window Help
input_block
Hier.1
Logical Hier
Cells (Hierarchical)
Report.1 - Area
Report.2 - Power
Report.3 - Timing
Hier.1
Report.1
Report.2
Report.3
Log
History
Options:
design_vision>
Ready
root@URANUS:~
constraints.tcl (~/PARSE...
work - File Browser
[pape_results (~) - gedit]
*Unsaved Document 1 - ...
Design Vision

Cell Name	Ref Name
packet_ar...	mem_32_1rp...
linked_list...	mem_32_1rp...
dest_head...	mem_8_1rp_...
dest_tail_...	mem_8_1rp_...
freelist	freelist_fifo
shift_reg	shift_reg_WI...

```
****************************************
Report : area
Design : input_block
Version: C-2009.06-SP5
Date   : Wed Mar  9 17:06:56 2011
****************************************

Library(s) Used:

    saed90nm_typ (File: /root/PARSER/ref/mc

Number of ports:                   58
Number of nets:                   524
Number of cells:                  374
Number of references:              21

Combinational area:       61719.417634
Noncombinational area:    74718.128324
Net Interconnect area:      undefined   (No

Total cell area:          136437.545957
Total area:                 undefined

***** End Of Report *****
```

```
Library(s) Used:

    saed90nm_typ (File: /root/PARSER/ref/


Operating Conditions: TYPICAL   Library:
Wire Load Model Mode: top


Global Operating Voltage = 1.2
Power-specific unit information :
    Voltage Units = 1V
    Capacitance Units = 1.000000pf
    Time Units = 1ns
    Dynamic Power Units = 1mW    (derived
    Leakage Power Units = 1pW


  Cell Internal Power  =   2.9435 mW   (9
  Net Switching Power  =  41.4356 uW
                        ---------
Total Dynamic Power    =   2.9849 mW   (10

Cell Leakage Power     = 315.9265 uW


***** End Of Report *****
```

```
linked_list_array/raddr[0] (mem_32_1rp_
linked_list_array/U136/Q (MUX41X1)
linked_list_array/U122/Q (MUX41X1)
linked_list_array/U121/Q (MUX21X1)
linked_list_array/rdata_tri[0]/Q (TNBUI
linked_list_array/rdata[0] (mem_32_1rp_
U412/Q (AO21X1)
dest_head_array/wdata[0] (mem_8_1rp_1wp
dest_head_array/U72/Q (AO22X1)
dest_head_array/data_reg[7][0]/D (DFFX1
data arrival time

clock my_sys_clk (rise edge)
clock network delay (ideal)
dest_head_array/data_reg[7][0]/CLK (DFI
library setup time
data required time
----------------------------------------
data required time
data arrival time
----------------------------------------
slack (MET)


***** End Of Report *****
```

```
data required time                                              19.93
-----------------------------------------------------------------------
data required time                                              19.93
data arrival time                                              -19.81
```

Fig. 4.1. Synthesis report for input block using standard cell library
(D flip flop memory)

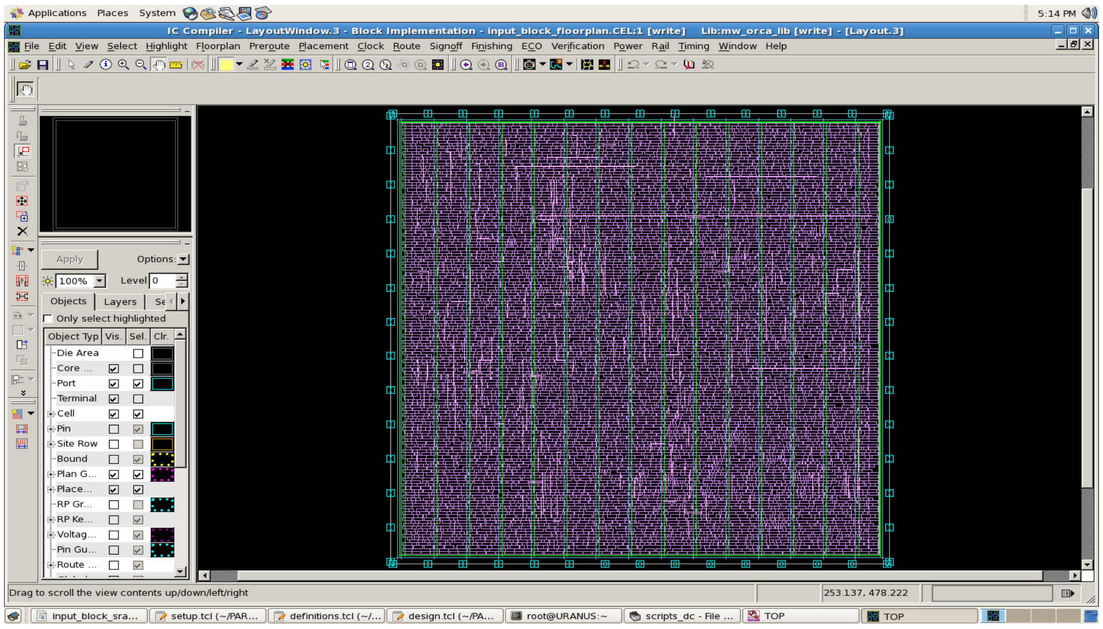
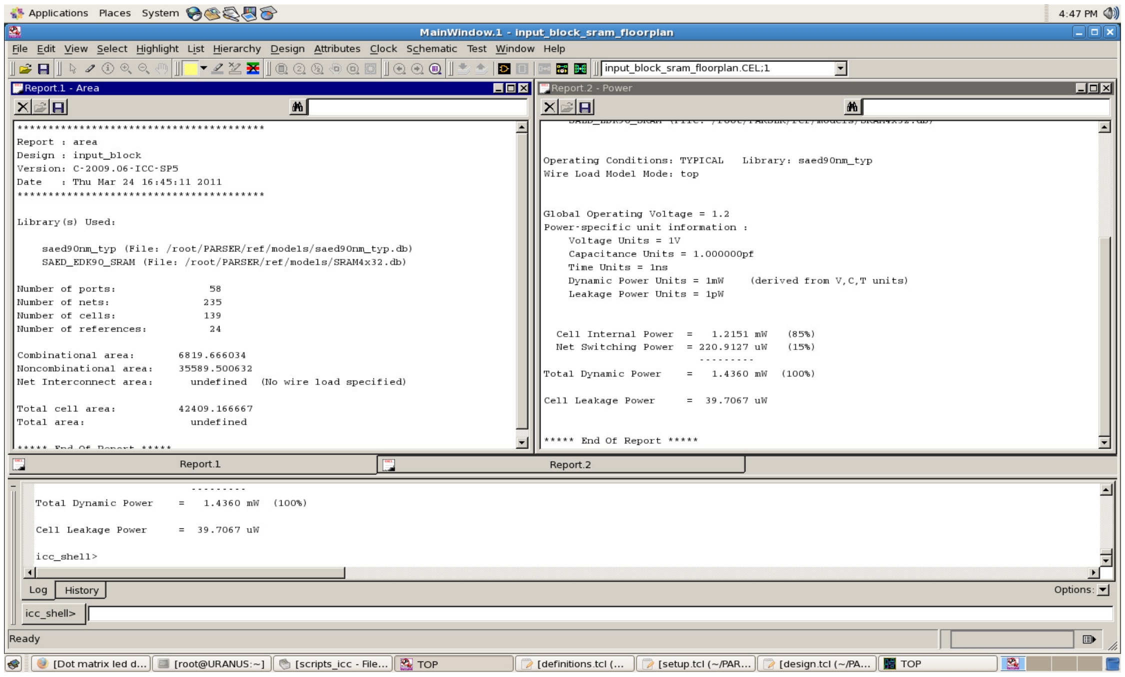

IC Compiler - LayoutWindow.3 - Block Implementation - input_block_floorplan.CEL;1 [write] Lib:mw_orca_lib [write] - [Layout.3]
File Edit View Select Highlight Floorplan Preroute Placement Clock Route Signoff Finishing ECO Verification Power Rail Timing Window Help
Apply
Options:
100%
Level 0
Objects Layers
Only select highlighted
Object Typ Vis. Sel. Clr.
Die Area
Core
Port
Terminal
Cell
Pin
Site Row
Bound
Plan G...
Place...
RP Gr...
RP Ke...
Voltag...
Pin Gu...
Route
Drag to scroll the view contents up/down/left/right
253.137, 478.222
5:14 PM

Fig. 4.2. Layout of input block using D flip flop memory

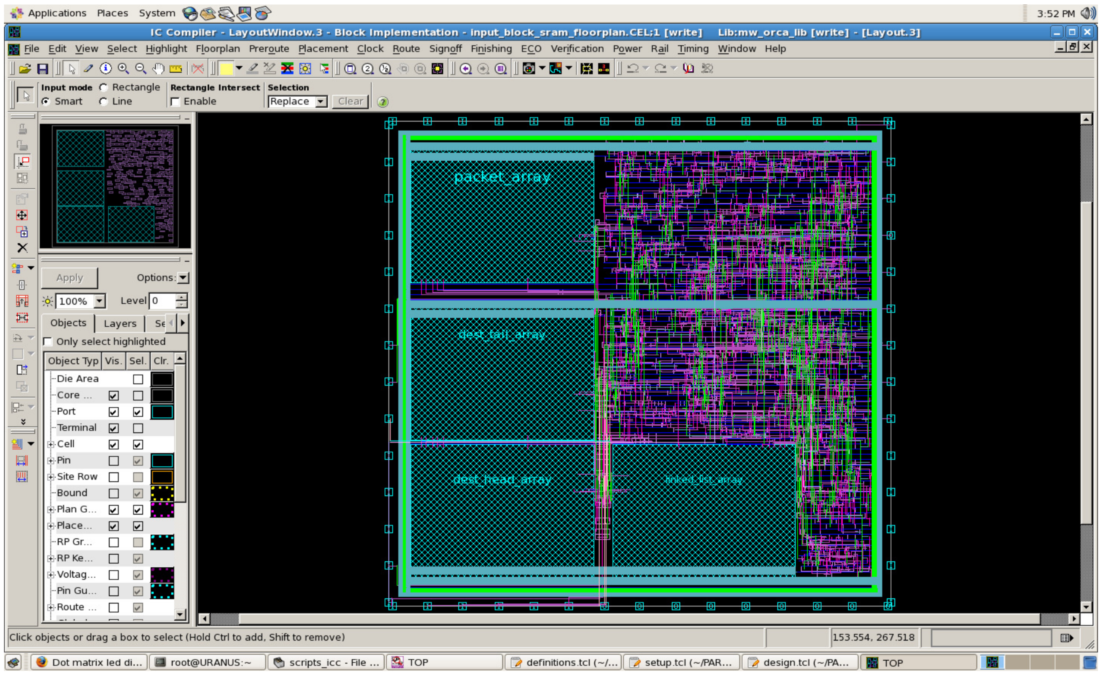

MainWindow.1 - input_block_sram_floorplan
File Edit View Select Highlight List Hierarchy Design Attributes Clock Schematic Test Window Help
input_block_sram_floorplan.CEL;1
Report.1 - Area
Report.2 - Power
Report.1
Report.2
Log History
Options:
Ready
4:47 PM

```
****************************************
Report : area
Design : input_block
Version: C-2009.06-ICC-SP5
Date   : Thu Mar 24 16:45:11 2011
****************************************

Library(s) Used:

    saed90nm_typ (File: /root/PARSER/ref/models/saed90nm_typ.db)
    SAED_EDK90_SRAM (File: /root/PARSER/ref/models/SRAM4x32.db)

Number of ports:                   58
Number of nets:                   235
Number of cells:                  139
Number of references:              24

Combinational area:        6819.666034
Noncombinational area:     35589.500632
Net Interconnect area:       undefined   (No wire load specified)

Total cell area:           42409.166667
Total area:                  undefined

***** End Of Report *****
```

```
Operating Conditions: TYPICAL   Library: saed90nm_typ
Wire Load Model Mode: top


Global Operating Voltage = 1.2
Power-specific unit information :
    Voltage Units = 1V
    Capacitance Units = 1.000000pf
    Time Units = 1ns
    Dynamic Power Units = 1mW    (derived from V,C,T units)
    Leakage Power Units = 1pW


  Cell Internal Power  =   1.2151 mW   (85%)
  Net Switching Power  = 220.9127 uW   (15%)
                         ---------
Total Dynamic Power    =   1.4360 mW  (100%)

Cell Leakage Power     =  39.7067 uW


***** End Of Report *****
```

```
                         ---------
Total Dynamic Power    =   1.4360 mW  (100%)

Cell Leakage Power     =  39.7067 uW

icc_shell>
```

Fig. 4.3. Synthesis report for input block using SRAM


IC Compiler - LayoutWindow.3 - Block Implementation - input_block_sram_floorplan.CEL;1 [write]   Lib:mw_orca_lib [write] - [Layout.3]

Click objects or drag a box to select (Hold Ctrl to add, Shift to remove)

153.554, 267.518

Fig. 4.4. Layout of input block using SRAM

4.1. Comparison chart showing results for area, power, and timing.

	Area	Power	Clock period
Input block (DFF)	136438 sq microns	2.985 mW	20 ns
Input block (SRAM)	42409 sq microns	1.436 mW	5 ns

Table 1

4.2. The results of the SIMULINK model for distributed and common buffer are as follows:

Simulation Model	Average Delay
Common Buffer (128 Packets)	4.3 time units
Distributed Buffers (4 x 32 Packets)	7.9 time units

Table 2

5. Conclusion:

The input block optimized for area and power by incorporation of custom SRAM blocks into the design. The custom SRAM blocks replace DFF(D flip flop) memory implementations. The amount of hardware resources required to store one bit of information using and SRAM cell (6 transistors) is much less than the hardware requirements for storing one bit of information using DFF (28 transistors).

The use of SRAM macros in place of standard cell D flip flops have resulted in an area reduction and corresponding reduction in power consumption. The improved design occupies approximately 30% of the area of the original design. This is in conformity to the ratio of the area of an SRAM cell to the area of a D flip flop, which is approximately 6:28. The power consumption is almost halved to 1.5 mW. Maximum operating frequency is improved from 50 Mhz to 200 Mhz.

The utilization efficiency of the packet buffer array improves when a common buffer is used instead of individual buffers in each input port. This is manifested in the form of lower delay in transferring a packet from the input to the output. The delay is improved by approximately 40% through the use of a common buffer.

REFERENCES :

[1] Salminen et al., survey of network-on-chip proposals, white paper, @ OCP-IP, march 2008.

[2] U. Ogras, J. Hu, and R. Marculescu, "Key Research Problems in NoC Design: A Holisitic Perspective," in Proceedings of the CODES, ISSS, 2005, pp. 69-74.

[3] R. Marculescu, U. Ogras, L. Peh, N. Jerger, and Y. Hoskote, "Outstanding Research Problems in NoC Design: System, Microarchitecture, and Circuit Perspectives," IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems, Vol. 28, No. 1, January 2009.

[4] E. Salminen, A. Kulmala, and T. H¨am¨al¨ainen, "On network-on-chip comparison," in Euromicro DSD, Aug. 2007, pp. 503–510

[5] S. Lee, "Implementation of a NoC", KAIST, 2005.

[6] E . Rijpkema, K. Goossens, A. Radulescu, J. Dielissen, J. van Meerbergen, P.Wielage, and E. Waterlander,"Trade-offs in the design of a router with both guaranteed and best-effort services for networks on chip", IEE Proc. on Computers and Digital Techniques, vol. 150, Issue 5, pp. 294-302, September 2003.

[7] A. Kumar, D. Manjunath, and J. Kuri, Communication Networking: An Analytical Approach, Morgan Kaufmann, 2004.

[8] P. T. Wolkotte, G. J. M. Smit, G. K. Rauwerda, and L. T. Smit, "An energy-efficient reconfigurable circuit- switched network-on-chip", Proc. 19th IEEE International Conference on Parallel and Distributed Processing Symposium, pp. 155-163, 2005.

[9] J. W. Dally and B. Towles, "Route packets, not wires: On-Chip interconnection networks", Proc. IEEE International Conference on Design and Automation, pp.684-689, June 2001.

[10] C. Albenes, Zeferino Frederico G. M. E. Santo, Altarniro Amadeu Susin, "ParlS: A parameterizable interconnect switch for Networks-on-Chips", Proc. ACM Conference, pp. 204-209, 2004.

[11] Y. Tamir and G. L. Frazier, "High-performance multi-queue buffers for VLSI communications switches," SIGARCH Comput. Archit. News, vol.16, no. 2, pp. 343–354, 1988.

[12] H. Obara, "Optimum architecture for input queuing ATM switches," Electronics Letters, vol. 27, no. 7, pp. 555–557, Mar. 1991.

[13] H. Obara and Y. Hamazumi, "Parallel contention resolution control for input queuing ATM switches," Electron. Lett., vol. 28, no. 9, pp. 838–839,Apr. 1992.

[14] Nick McKeown, "The iSLIP scheduling algorithm for input-queued switches," IEEE/ACM Trans. Netw., vol. 7, no. 2, pp. 188–201, 1999.

[15] Rehan Maroofi, V. N. Nitnaware and Dr. S. S. Limaye, "AREA-EFFICIENT DESIGN OF SCHEDULER FOR ROUTING NODE OF NETWORK-ON-CHIP" International Journal of VLSI design & Communication Systems (VLSICS) Vol.2, No.3, September 2011.

[16] Adnan Aziz, Amit Prakash, and Vijaya Ramachandra, "A near optimal scheduler for switch-memory-switch routers," in Proceedings of the fifteenth annual ACM symposium on Parallel algorithms and architectures, SPAA '03, 2003, pp. 343–352.

[17] A semester project of John D. Pape on "Implementation of an On-chip Interconnect Using the iSLIP Scheduling Algorithm" at The University of Texas at Austin December 11, 2006

[18] SYNOPSYS Manual for Design Compiler, IC Compiler, available at SOLVNET {https://solvent.synopsys.com/}.