Evaluation of FPGA-based Implementations of Interconnection Networks

Master Thesis Presentation

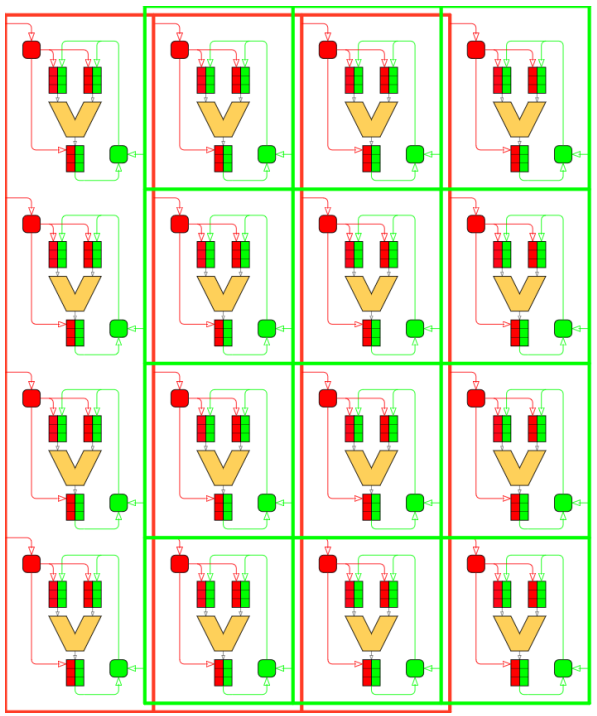
Carsten Harms

University of Kaiserslautern, Embedded Systems Group

July 28, 2016

SCAD Architecture

- Synchronous Control Asynchronous Dataflow (SCAD)
  - Only *MOVE* instructions via MIB
  - 32 Functional Units (FU) connected to each other via DTN
  - Allows for arbitrary latencies in data path
  - Uses no registers, only in-/output queues



Source: es.cs.uni-kl.de

Functional Unit

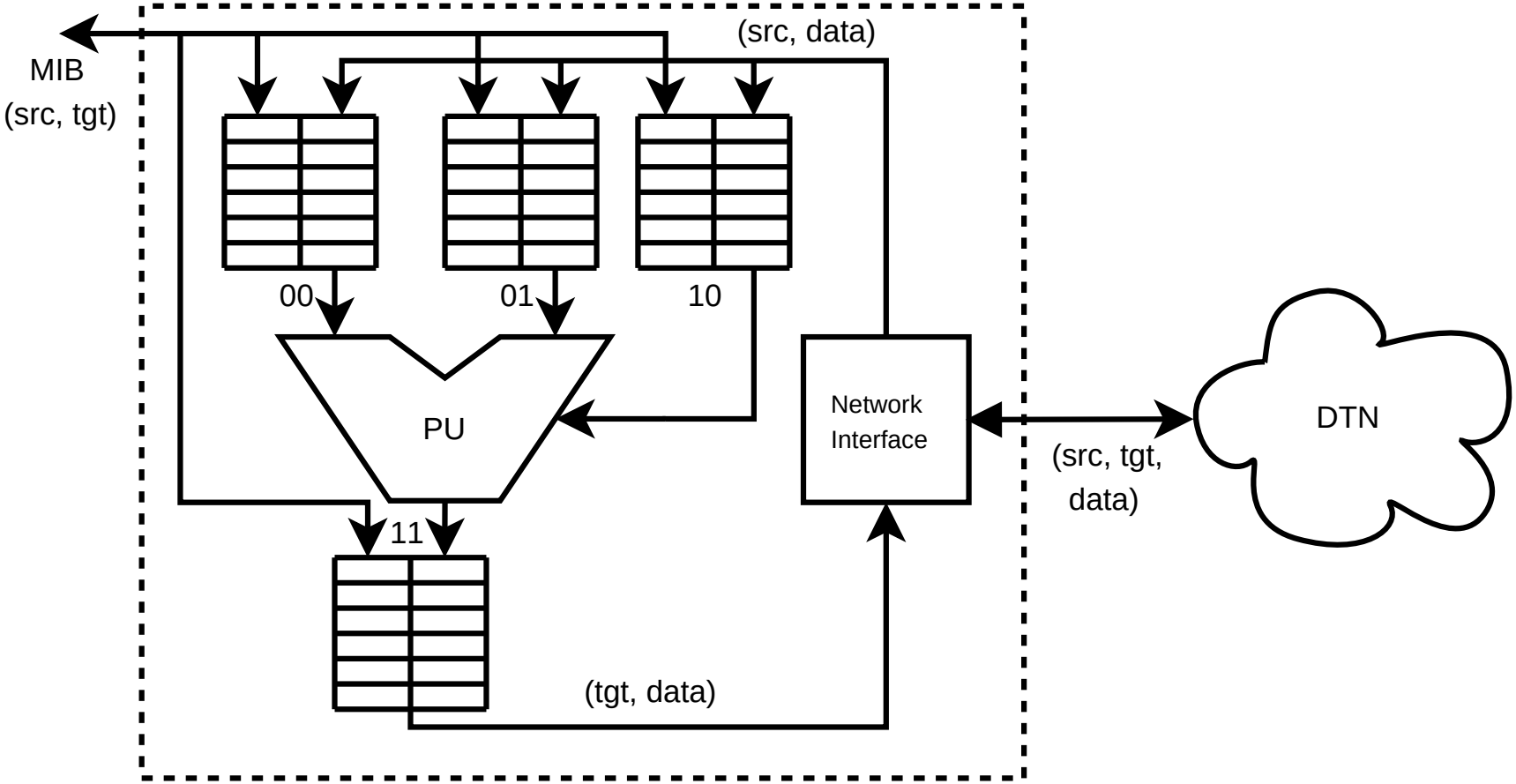
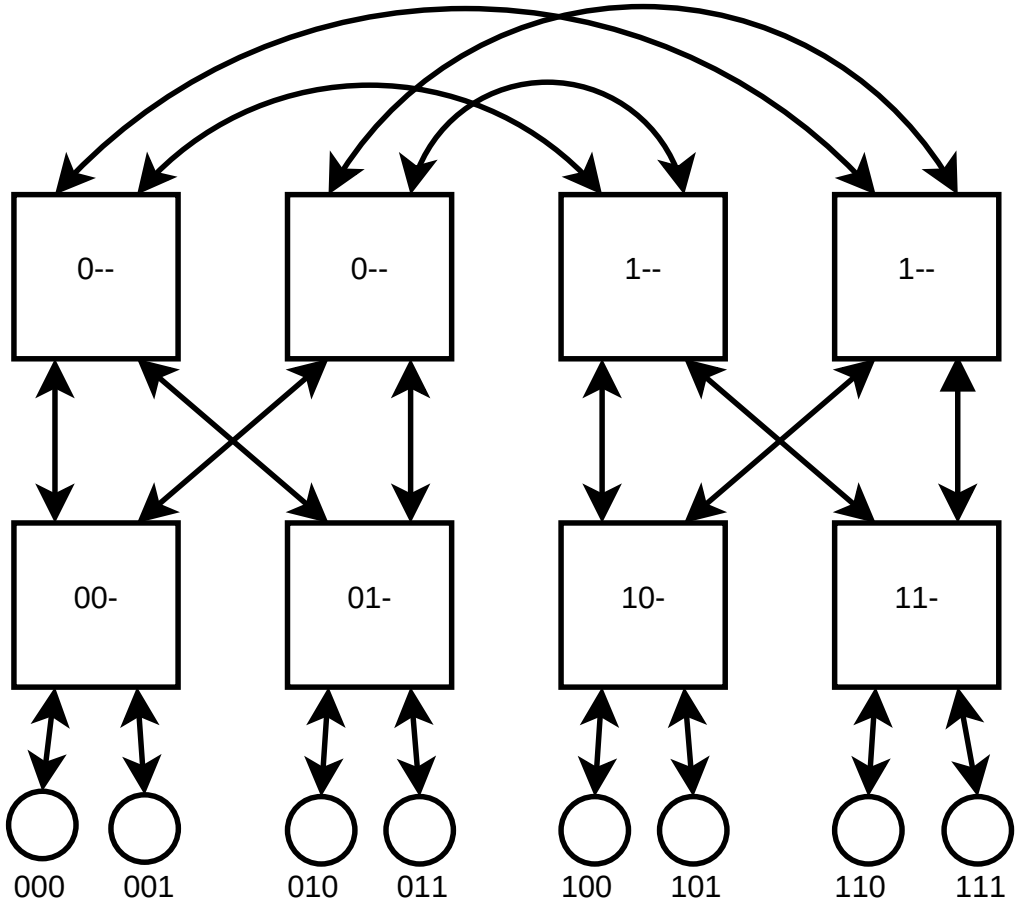


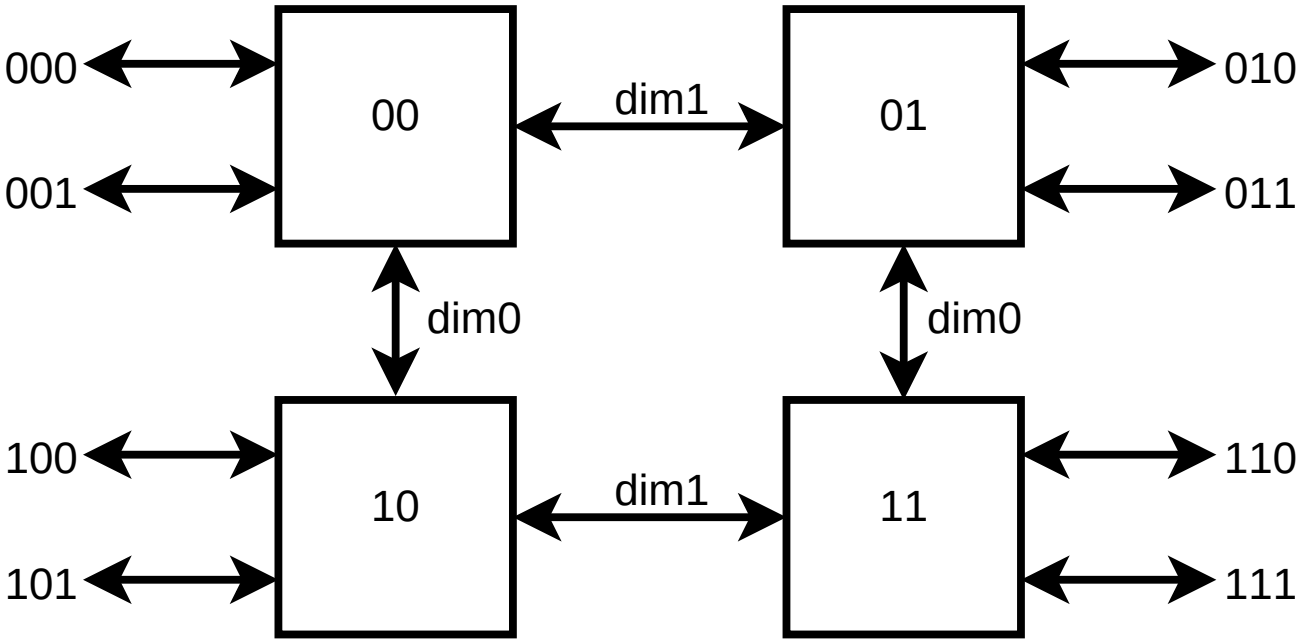
Figure: Basic block diagram of a complete functional unit with three input ports 00, 01 and 10 and a single output port 11. Only data flow and MIB are shown, other signals omitted for readability.

Folded Clos (Fat-Tree) Topology



- Many, but very simple routers → higher minimal # of hops
- Routing: Compare *significant* router address bits with destination

Flattened Butterfly Topology



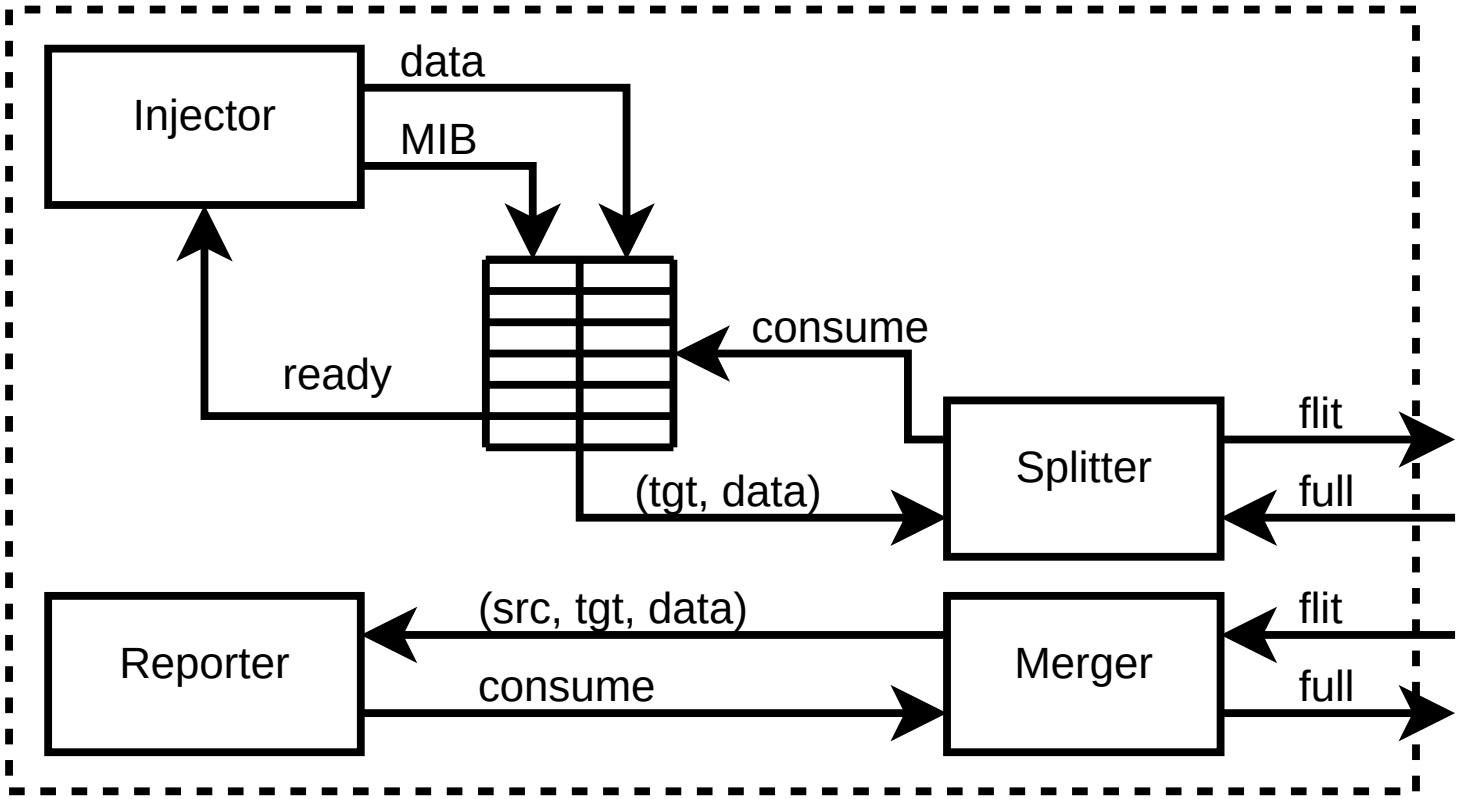
- Less, but increasingly complex routers → lower minimal # of hops
- DOR: Compare dimensions one by one in fixed order, route towards mismatches (oblivious minimal routing)
- VAL: Randomly select intermediate router, route via other algorithms. Needs at least two virtual channels.
- Greedy: Needs at least # of inter-router dimensions virtual channels.

Network Specifics

- Requirements: dead and livelock-free, fair & no packet loss
- Wormhole-Routing:
  - Lowest delay and lowest memory demands
  - Prone to deadlocks
- Message Format, equally sized flits:

#	flit[7:3]	flit[2:1]	flit[0]
1	target.unit	target.queue	0
2	source.unit	source.queue	0
3	Data[31:24]		
4	Data[23:16]		
5	Data[15:8]		
6	Data[7:0]		

Simulated Functional Unit



- Probabilistic injection
- Store current cycle number in data field
- Finite buffer size (6 entries)

Traffic Patterns

- Bit permutations:

Pattern	Source Address	Target Address
Bit Complement	$(b_4, b_3, b_2, b_1, b_0)$	$(\neg b_4, \neg b_3, \neg b_2, \neg b_1, \neg b_0)$
Bit Reverse	$(b_4, b_3, b_2, b_1, b_0)$	$(b_0, b_1, b_2, b_3, b_4)$
Bit Rotation	$(b_4, b_3, b_2, b_1, b_0)$	$(b_3, b_2, b_1, b_0, b_4)$
Bit Shuffle	$(b_4, b_3, b_2, b_1, b_0)$	$(b_0, b_4, b_3, b_2, b_1)$
Bit Transpose	$(b_4, b_3, b_2, b_1, b_0)$	$(b_1, b_0, b_2, b_4, b_3)$

- Unified Random

Methodology

- Xilinx Vivado 2015.3 VHDL simulator
- Testbench, three Phases:
  - Warm-Up (1000 cycles) to steady-state
  - Measurement (5000 cycles)
  - Drain (500 cycles)
- Latency (Measurement & Drain):

$$Latency = cycle_{start} - cycle_{current}$$

- Throughput:
  - Number of messages during measurement phase

Full-Load Performance Comparison

- Fat-Tree (flit size = 16):

Traffic Pattern	Latency [cycle] Average	Latency [cycle] Maximum	Absolute Throughput
Bit Complement	43	57	22760
Bit Reverse	140	169	5715
Unified Random	65	152	15647

- Flattened Butterfly (flit size = 16):

Traffic Pattern	Latency [cycle] Average	Latency [cycle] Maximum	Absolute Throughput
Bit Rotation	99	254	7901
Bit Shuffle	38	57	19398
Unified Random	52	90	15761

Impact of Flit-Size under Full Load

Fat-Tree:

Traffic Pattern	Flit Size	Latency [cycle] Average	Latency [cycle] Maximum	Absolute Throughput
Worst-case	16	140	169	5715
Worst-case	8	172	210	4000
Unified-Random	16	65	152	15647
Unified-Random	8	91	203	10022

Flattened Butterfly:

Worst-case	16	99	254	7901
Worst-case	8	146	347	5172
Unified-Random	16	52	90	15761
Unified-Random	8	75	125	10725

Impact of Network Size under Full Load

Size	Latency Avg.	Max.	TP
32	140	169	5715
16	108	129	3335
8	27	42	3237

(a) Fat-Tree on WC pattern

Size	Latency Avg.	Max.	TP
32	65	152	15647
16	60	129	7745
8	35	69	5467

(b) Fat-Tree on UR pattern

Size	Latency Avg.	Max.	TP
32	99	254	7901
16	79	214	4545
8	55	71	2727

(c) Flattened Butterfly on WC pattern

Size	Latency Avg.	Max.	TP
32	52	90	15761
16	47	82	8429
8	37	66	4937

(d) Flattened Butterfly on UR pattern

Fat-Tree Resource Utilization

- Post-synthesis resource utilization:

Net.	Flit	LUTs	Flip-Flops
8	8	564 (1.06%)	560 (0.53%)
8	16	616 (1.16%)	784 (0.74%)
16	8	1744 (3.28%)	1680 (1.58%)
16	16	1944 (3.65%)	2352 (2.21%)
32	8	4400 (8.27%)	4480 (4.21%)
32	16	5104 (9.59%)	6272 (5.89%)

Flattened Butterfly Resource Utilization

- Post-synthesis resource utilization:

Net.	Flit	LUTs	Flip-Flops
8	8	629 (1.18%)	368 (0.35%)
8	16	708 (1.33%)	480 (0.45%)
16	8	2050 (3.85%)	887 (0.83%)
16	16	2532 (4.76%)	1167 (1.10%)
32	8	5042 (9.48%)	2112 (1.98%)
32	16	6538 (12.29%)	2784 (2.62%)

Flattened Butterfly with Virtual Channels

- Network size of 32 terminals
- Post-synthesis resource utilization with two virtual channels using VAL & DOR within:

Flit Size	LUTs	Flip-Flops
8	15912 (29.91%)	5120 (4.81%)
16	20041 (37.67%)	6240 (5.86%)

Conclusion

- Doubling bandwidth increases throughput roughly 40–60% and lowers latency by roughly 20–30%
  - Resource utilization increases only up to 30%
- Flattened butterfly significantly less latency on UR than fat-Tree
  - Job of compiler to partition code at least randomly to FUs
- Fat-tree uses least amount of critical resources (LUTs)
  - Flattened butterfly for 32 terminals uses 28% more LUTs
- Virtual channels for 32 node flattened butterfly too expensive
  - Adaptive and/or non-minimal routing unfeasible

Time for Questions

Thank you for your attention!