ZUMA: An Open FPGA Overlay Architecture

Alex Brant, Guy G.F. Lemieux

Department of Electrical and Computer Engineering

University of British Columbia

ZUMA Overlay

- An Embedded FPGA (eFPGA), aka "FPGA-on-an-FPGA"
  - Open, cross-compatible architecture
  - Open architecture for open P&R tools
- Compiled onto Xilinx and Altera FPGAs
- Place and route using VTR

Motivation

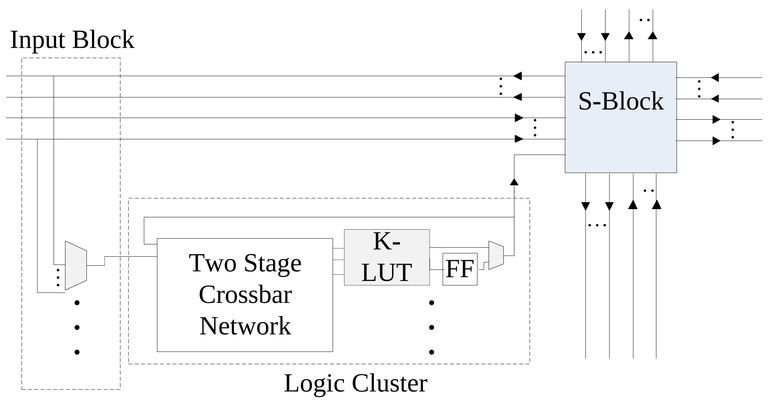
- Enabling Applications
  - Dynamic reconfiguration on any device
  - User-configured logic added to fixed FPGA bitstream
  - Bitstream compatibility between vendors and parts
- Enabling Research
  - Open access to architecture & CAD
  - Device-independent platform for reconfigurable computing

Contributions

- An open FPGA with fully disclosed details
  - Adds flexibility and capability to real FPGAs
  - Efficient implementation of LUTs and MUXs on real FPGAs
  - Target for new, open tools

Architecture

- Similar to classic VPR architecture
- Single-driver routing for efficient implementation on FPGA host
- Implemented in two ways
  - Generic Verilog (compiles to any FPGA)
  - FPGA-specific Verilog (compiles efficiently to specific FPGA)

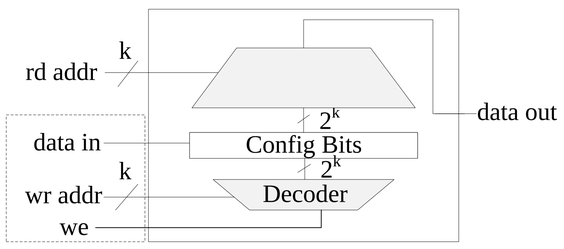


One ZUMA Tile

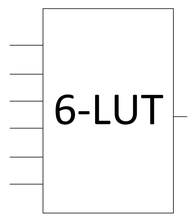
Details

- Generic implementation overheads
  - To build one 6-eLUT, requires 21 6-LUTs for the 64:1 mux alone
  - To build one 16:1 mux, requires five 6-LUTs
  - Plus many FFs needed for configuration bits
- More efficient using LUTs in "LUTRAM" mode
  - To build entire 6-eLUT, requires just one 6-LUT in RAM mode
    - Saves twenty 6-LUTs
  - To build one 16:1 mux, requires three 6-LUTs in RAM mode
    - Saves two 6-LUTs (see figure in next column)

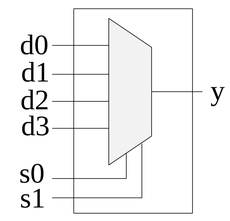
Details (continued)



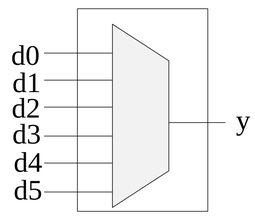
K-input LUT in RAM Mode



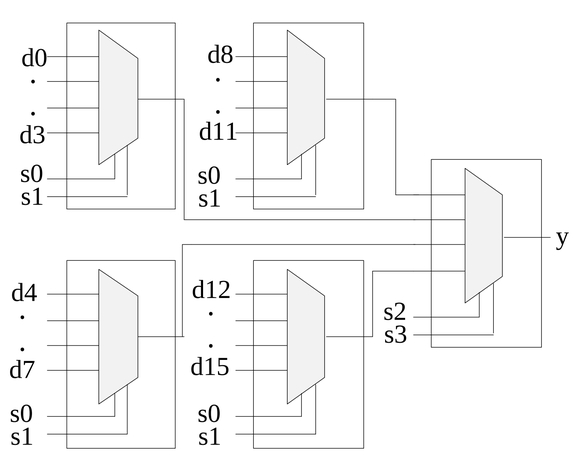
6-LUT



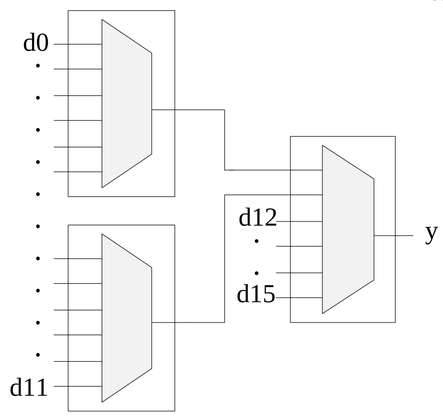
6-LUT, configured as a 4-to-1 MUX



6-LUT, configured as a 6-to-1 MUX in RAM mode

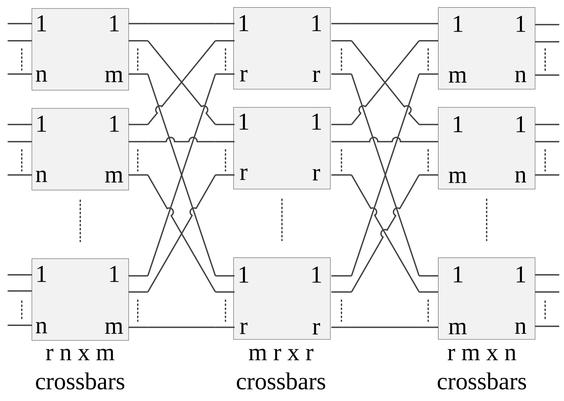


16-to-1 MUX built using five 6-LUTS

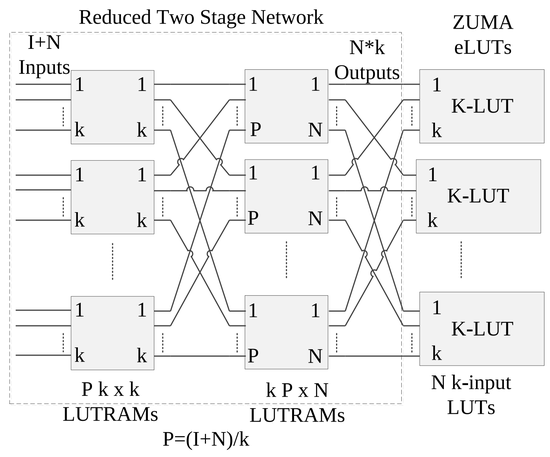


16-to-1 MUX built using three 6-LUTs in RAM mode

- Fully connected logic cluster interconnect (LUT input selection matrix) is large
  - Replace with Clos network, routes any input to any output
  - Clos network is 3 stages, but 3rd stage can be LUTs themselves
  - Crossbars in Clos network built from eMUXs using LUTs in RAM mode



Generic Clos Network



Two-Stage Crossbar Network used in ZUMA Logic Cluster

Results

- We have built Xilinx and Altera versions
  - Using 4-LUT (Spartan/Cyclone) and 6-LUT (Virtex/Stratix) hosts
- VTR to place and route 19 MCNC benchmarks
  - Using 6-LUTs, N=8, L=4 wires, requires channel width of 112
  - Area overhead as low as 40 LUTs per eLUT
  - Our generic version is 125-150 LUTs per eLUT

	Generic Verilog		Custom Xilinx Version	
	Host LUTs	Host FFs	Host LUTs	Host FFs
Switch Block	121	156	104	0
Input Block	56	288	56	0
Crossbar	**288**	248	**144**	0
BLEs	**528**	520	**16**	8
Total	993	1212	320	8
Total *per eLUT*	124.1	151.5	40	1

Xilinx Virtex 5 used as host for both Generic and Custom versions.

Altera host is similar, but uses more FFs.

- Prior work: Virtual FPGA architecture 3-LUTs, N=4, channel width of 32
  - Implementation 1 uses 86 4-LUTs + 554 FFs per eLUT
  - Implementation 2 uses 354 4-LUTs + 234 FFs per eLUT

R. Lysecky et al., “Firm-core Virtual FPGA for Just-in-Time FPGA Compilation,” poster at FPGA2005

IEEE Symposium on Field-Programmable Custom Computing Machines 2012

This research has been funded by NSERC CRSNG