Overlay Architectures for Space Applications

L. Fiolhais, F. Gonçalves
R. Martins, J.T. de Sousa

jose.sousa@ipbloq.com

Outline

- Motivation
- Introduction
- Architecture
- Toolchain
- Applications
- Results
- Conclusion

Motivation

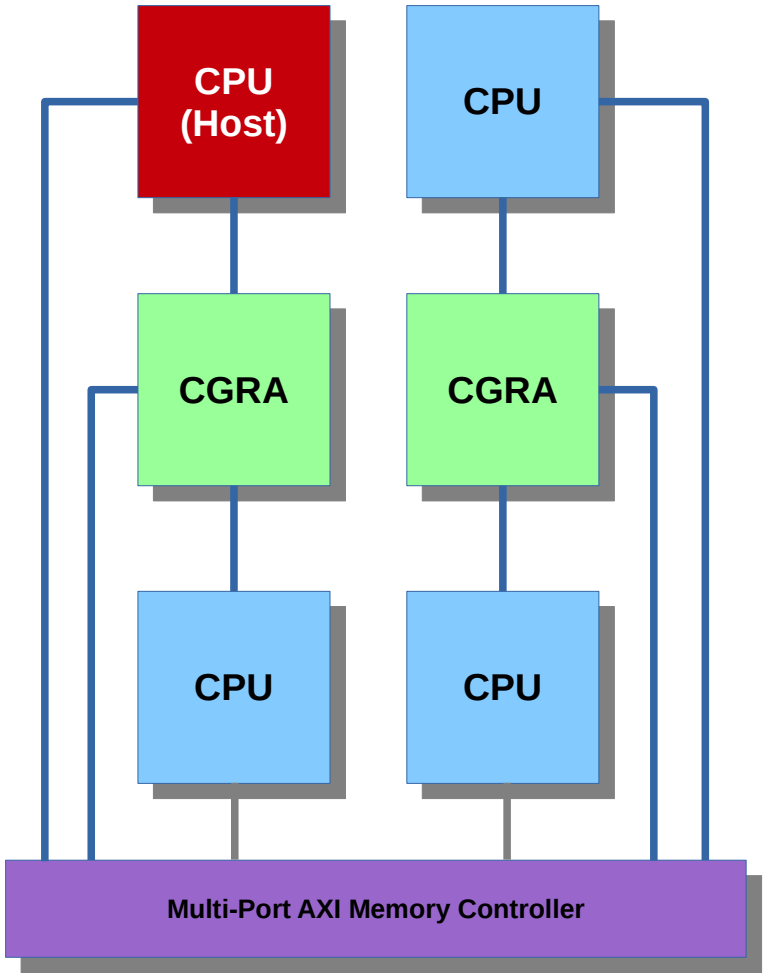
Current heterogeneous systems could be better optimized for silicon area, performance and power consumption

- **CPUs** spend more hardware on efficient instruction handling than in computation
- **FPGAs** are used for mapping arbitrary circuits, spending more hardware on configuration infrastructure than in computation
- **GPUs** have many generic stream processors to brute force highly parallel problems using arrays of simple CPUs

Introduction

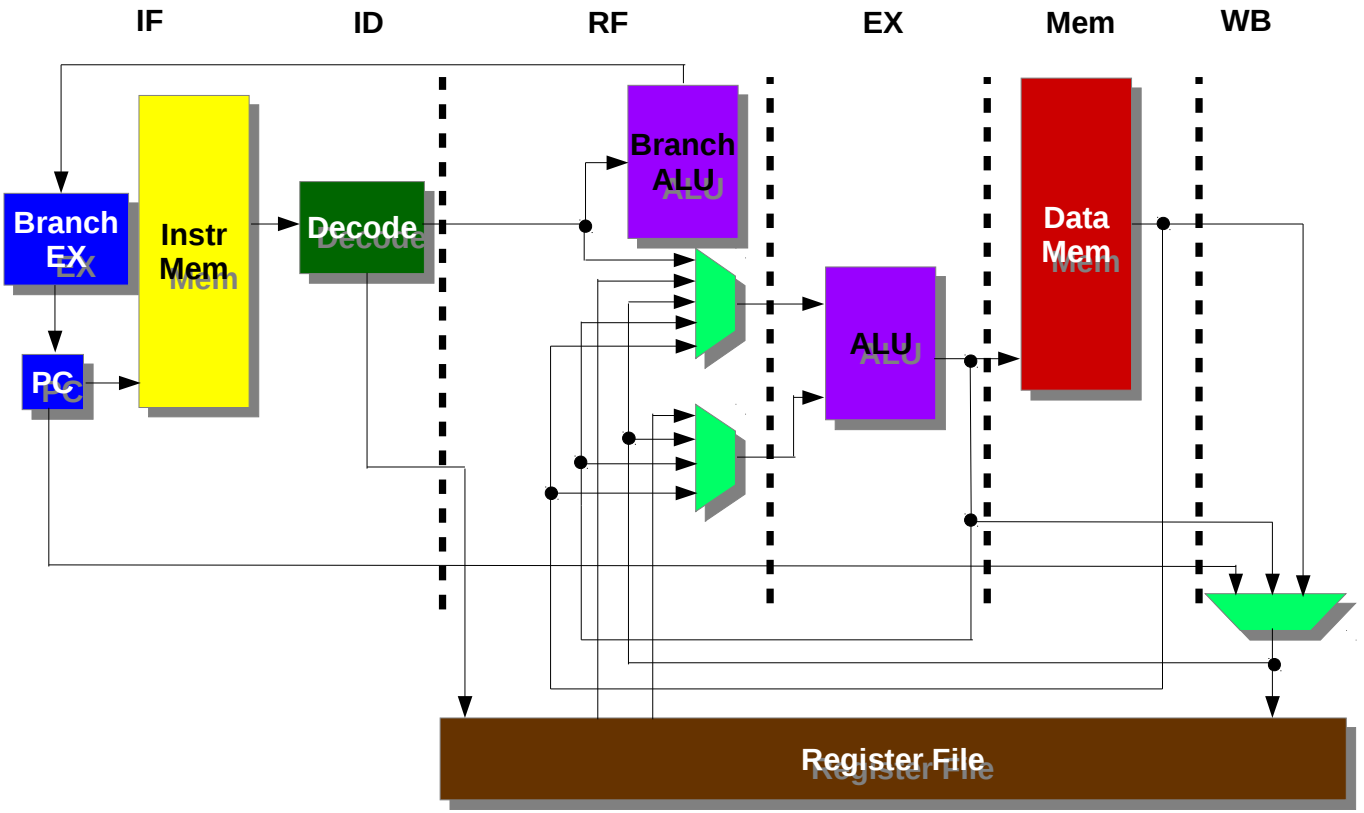
- **New architecture**: a few simple CPUs combined with a more suitable type of reconfigurable hardware, the *Coarse-Grained Reconfigurable Array (CGRA)*
- The **CPU** can be any proven one such as *LEON*, or a more innovative one such as the new and open *RISC-V ISA*. Currently the *Fireworks* CPU developed by parent company Coreworks SA is being used
- The **CGRA** used is the patented and silicon proven *SideWorks CGRA* also developed by CoreWorks SA
- New architecture targets low power systems which require a high degree of performance

Architecture overview



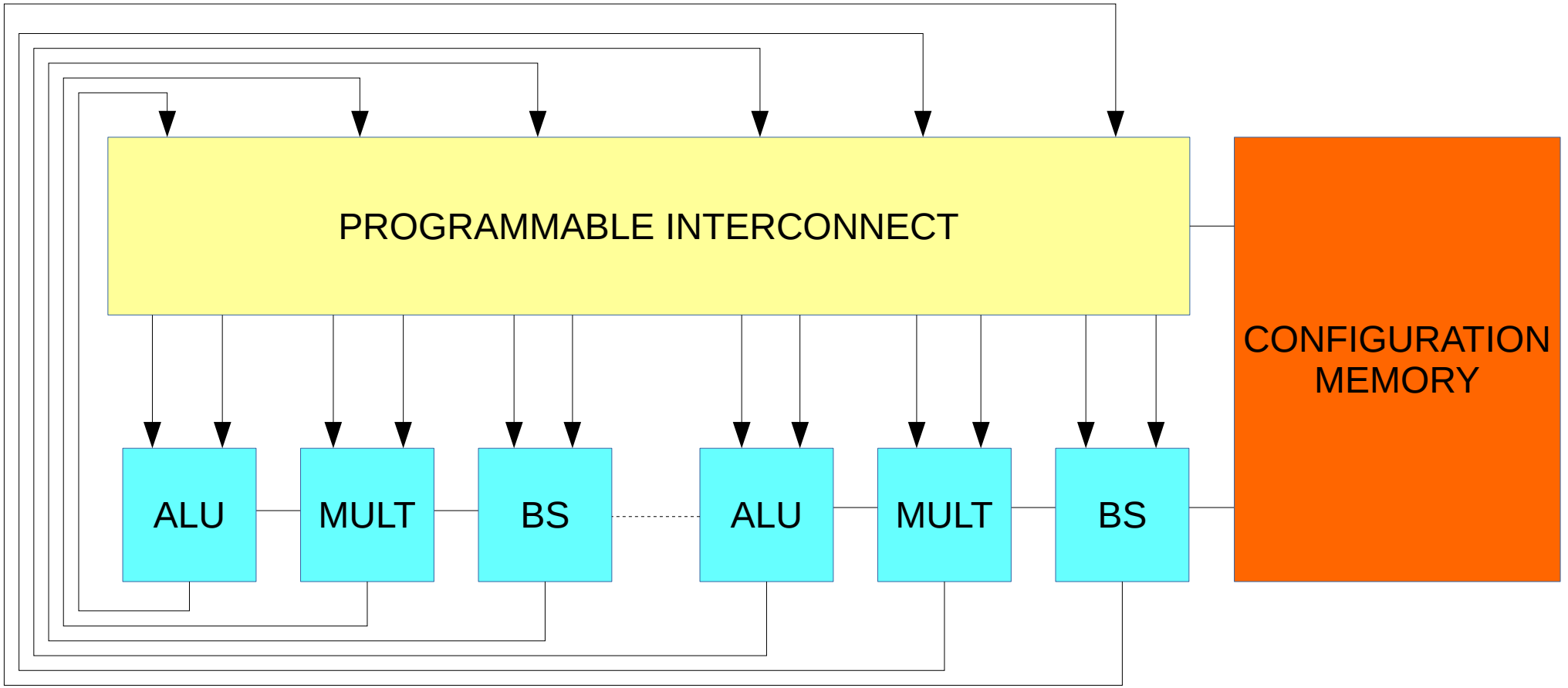
- Core interconnection uses AHB/APB/AXI
- CPUs are masters
- CGRAs are slaves and arbitrate using round-robin scheme
- All cores access the external memory through a master interface

Architecture – Fireworks CPU



- 5-Stage Pipeline
- Integer instructions only
- 1.32 DMIPS of Performance
- RISC-V evolution planned
- Much smaller than LEON3 (4x)

Architecture – SideWorks CGRA



FPGA versus CGRA

	FPGA	CGRA
PROGRAMMABLE HARDWARE?	YES	YES
DATA WIDTH	BIT	WORD
BASIC BLOCK	LUT	OPERATOR (+,-,*,/,&,|,^, etc)
CREATES	CIRCUITS	DATAPATHS
CONFIGURATION OVERHEAD	LARGE	SMALL
RECONFIGURATION TIME	SLOW (ms:full, us:partial)	VERY FAST (ns)
FLEXIBILITY	HIGH	MODERATE
POWER CONSUMPTION	HIGH	LOW
CREATES	CIRCUITS	DATAPATHS

Overlay Architecture

- Implement CGRA in FPGA
  - Programmable circuit in a programmable circuit
  - 2 levels of programmability
    - FPGA level – slow reconfiguration
    - CGRA level – very fast reconfiguration
  - Portability like a virtual machine
  - Virtualization of large datapaths
    - If it does not fit, break it in parts and reconfigure quiclky

Toolchain – Fireworks

- Uses the GNU toolchain for compiling, debugging and profiling
- Compiler is **GCC** and debugger is **GDB**, the same ones you have on your Linux computer
  - Currently, Fireworks doesn’t support GDB
  - Debugging is done through standard IO or in extreme cases using RTL simulation

Toolchain – Sideworks

- **SideGUI:** GUI for entering the hardware datapaths
- **SideGen:** produces a Verilog Sideworks instance capable of executing all specified datapaths and more
- **SideConf:** programs datapaths onto the existing instance or reports mapping failure
- **SideSim:** fast RTL simulator based on the open source Verilator simulator (converts Verilog into a C++ object exercised with a SystemC testbench).

FPGA Results – Fireworks CPU

Implementation Results for Intel Arria V

Logic (ALMs)	#Regs	RAM (kbit)	#DSP	Fmax (MHz)
1,675	1,470	266	4	120

FPGA Results – Fireworks CPU

Implementation Results for Xilinx Kintex 7

Logic (LUTs)	#Regs	RAM (kbit)	#DSP	Fmax (MHz)
2,533	1,405	266	4	90

ASIC Results – Fireworks CPU

Implementation Results for TSMC

N(nm)	A(mm²)	M(kB)	F(MHz)	P(mW)
65	0.92	32	420	58
40	0.35	32	683	36
28	0.17	32	975	25

Results – Sideworks CGRA Instances

Instance	#ALU	#MUL	#BarrelShift	#Mem
1	6	4	1	4
2	14	5	5	5
3	8	4	4	5
4	8	8	0	5

FPGA Results – Sideworks CGRA

Implementation Results for Xilinx Kintex 7

Instance	Logic (LUTs)	#Regs	RAM (kbit)	#DSP	Fmax (MHz)
1	12,510	4,396	360	16	102
2	23,750	13,158	630	25	90
3	21,184	11,740	630	21	90
4	16,647	11,699	630	36	90

FPGA Results – Sideworks CGRA

Implementation Results for Intel Arria V

Instance	Logic (ALMs)	#Regs	RAM (kbit)	#DSP	Fmax (MHz)
1	8,607	4,673	351	32	130
2	15,340	12,620	623	25	120
3	13,700	11,270	623	21	120
4	10,680	11,200	623	36	120

ASIC Results – Sideworks CGRA

Implementation Results for TSMC

Instance	N(nm)	A(mm²)	M(kB)	F(MHz)	P(mW)
1	130	5.20	46	170	132
1	40	0.49	46	553	41
2	65	2.38	76	500	179
2	40	0.90	76	813	110
2	28	0.44	76	1161	77
3	65	2.18	76	510	167
3	40	0.83	76	829	102
3	28	0.41	76	1184	72
4	65	2.10	68	510	160
4	40	0.70	68	829	86
4	28	0.34	68	1184	60

Results vs. ARM Cortex A9

Core (40 nm) / Software	Area (mm^2)	RAM (kB)	Freq. (MHz)	Power (mW)	Speed Up
ARM Cortex-A9 / 1024-point FFT	4.60	64	800	500	1
ARM Cortex-A9 / K-means	4.60	64	800	500	1
ARM Cortex-A9 / HE AAC 5.1 Decoder	4.60	64	50	31	1
ARM Cortex-A9 / HE AAC 5.1 Encoder	4.60	64	70	44	1
1xSiw-1 / 1024-point FFT	0.49	46	553	41	18
1xSiw-1 / K-means Clustering	0.49	46	553	41	3.8
3xFiw + Siw-4 / HE AAC 5.1 Decoder	1.75	164	80	21	1
3xFiw + Siw-2 / HE AAC 5.1 Encoder	1.95	172	80	23	1

Conclusion

- The Fireworks CPU possesses a competitive energy footprint and resource usage
- The Sideworks CGRA allows a large set of applications to be executed with minimal energy requirements and high performance
- The presented system compared to an ARM Cortex-A9 system equipped with the NEON unit:
  - Can be much smaller (10x) and greener (>10x)
  - **FPGA/Silicon proven architecture with happy customers for almost 2 decades**