A VLIW Processor with Hardware Functions: Increasing Performance While Reducing Power

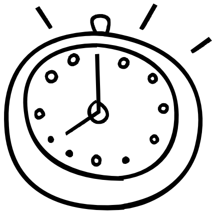
Profs. Ray Hoare and Alex Jones

Dara Kusic, Joshua Fazekas, Gayatri Mehta, and John Foster

Objective

- High Performance
  - □ PC to Supercomputer Performance
- Low Power
  - □ PDA Power

- Rapid Design Cycle
  - □ "Compile Software into Hardware"
  - □ Unmodified C Source Code
  - □ No Hardware Design Experience Required
  - □ Automatic Integration with Processor

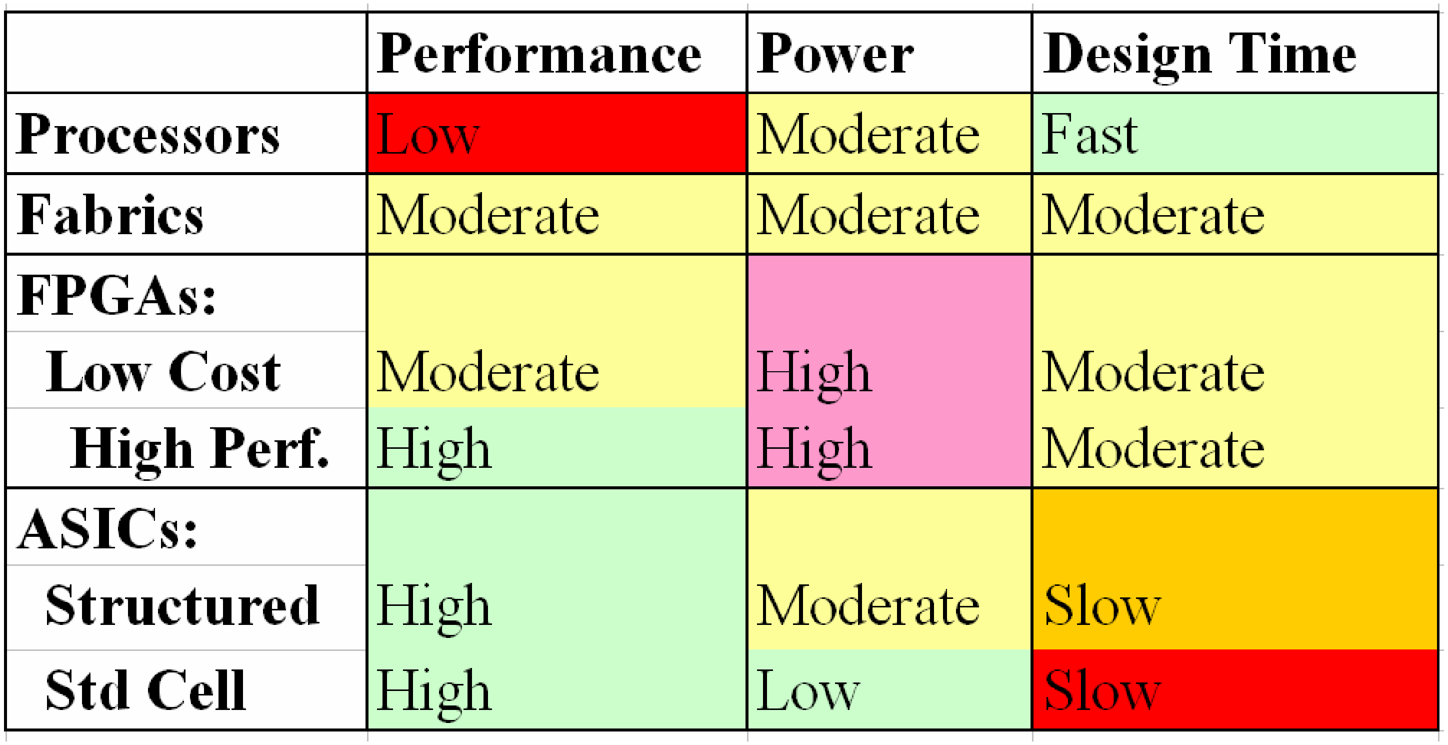


Target Technologies

For all devices, we want to

Maximize Performance while

Minimizing Power consumption and Design Time.



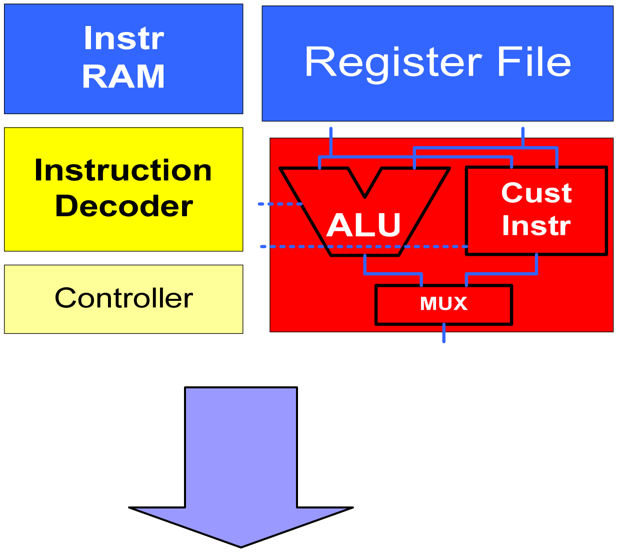
	Performance	**Power**	**Design Time**
Processors	Low	Moderate	Fast
Fabrics	Moderate	Moderate	Moderate
FPGAs:			
Low Cost	Moderate	High	Moderate
High Perf.	High	High	Moderate
ASICs:			
Structured	High	Moderate	Slow
Std Cell	High	Low	Slow

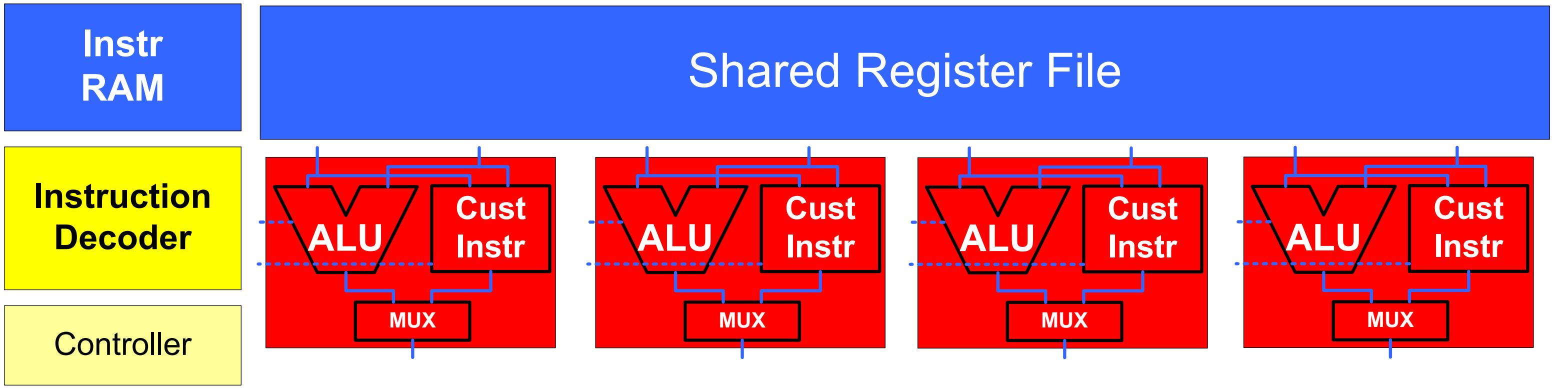
Our Approach

- VLIW Architecture
  - Up to 4x performance improvement for software alone
  - Full 32-bit Processor
    - Real applications
    - Altera NIOS II Instruction Set Architecture
- “Hardware Functions”
  - Convert critical software kernels into hardware
  - **9x to 332x Performance improvement**
  - **42x to 418x Power reduction**
- **Productivity** through Design Automation
  - Source: Unmodified C programs
  - **“Compile” software into hardware functions**

VLIW Processor: Exploit Instruction Level Parallelism

- Full NIOS II 32-bit ISA
  - □ Real processor, real apps
- Expanded to 4 ALUs for up to 4x speedup
  - □ Shared Register File
  - □ Shared Instruction Stream
  - □ Trimaran Compiler



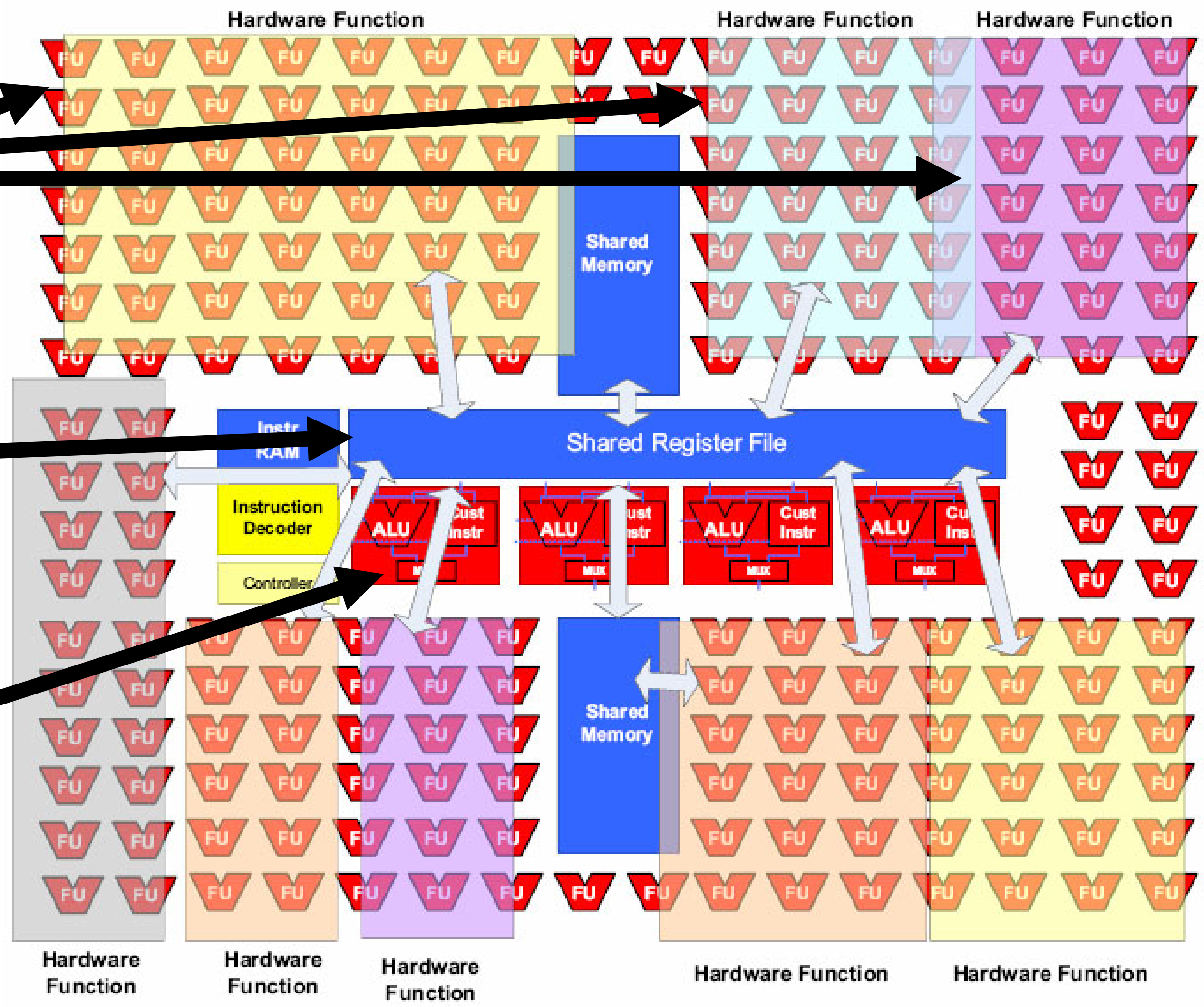


Hardware Functions:
Exploit Hardware Acceleration

Kernels converted to Hardware

Data shared in Reg. File

Control flow in Software



VLIW Exploits Instruction Parallelism Hardware Provides Acceleration

Kernels converted to Hardware → High Performance and Low Power

Data shared in Reg. File → No Data movement

Control flow in Software → Programmable

Benchmark Codes

- Speech Coding
  - ADPCM Encoder (4:1 compression)
  - ADPCM Decoder
  - G.721 (cordless phones)
  - GSM (wireless phones, 8:1 compression)
- Communications
  - Spherical Decoder (M tx, N rx antennas)
    - (Written in vectorized Matlab)
- Multimedia Decoding
  - MPEG2 Decoder (IDCT)

Performance Results

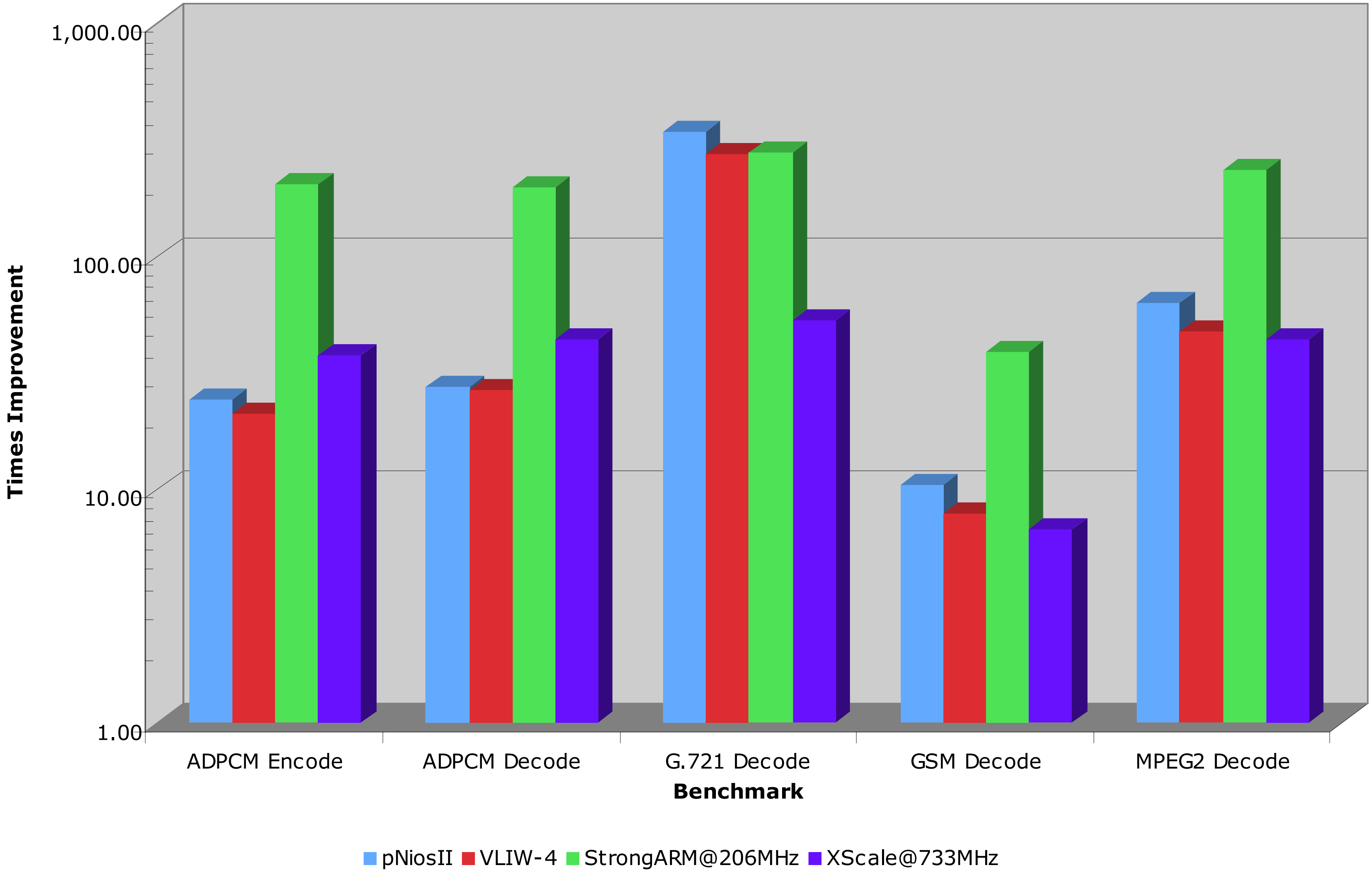
- Fully Implemented Processors
  - FPGA Comparison: Soft-core processor vs. HW Function in 90nm FPGA
  - **pNIOS:** our sequential processor
  - **VLIW-4:** our 4-wide VLIW
- Hardware Functions vs. Equivalent Software
  - Estimations based on a shared register file with HW Functions
  - **Intel Strong ARM 206MHz vs 160nm ASIC HW Functions**
  - **Intel XScale 733MHz vs 160nm ASIC HW Functions**

Performance Results

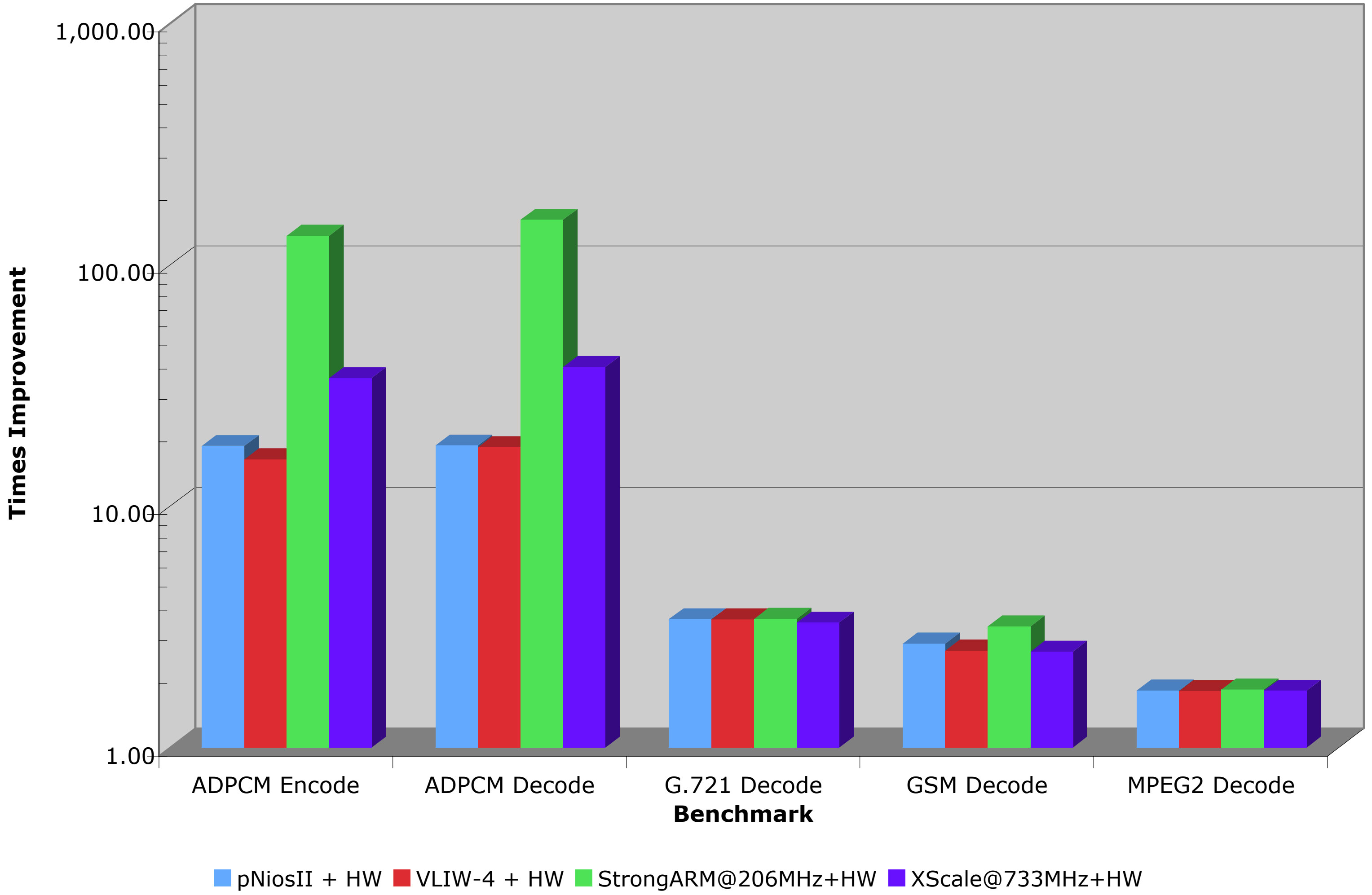
Acceleration Perspective

- A speedup of 2x turns 100MHz system into 200MHz of performance
- 10x provides 1GHz performance
- 100x provides 10GHz performance

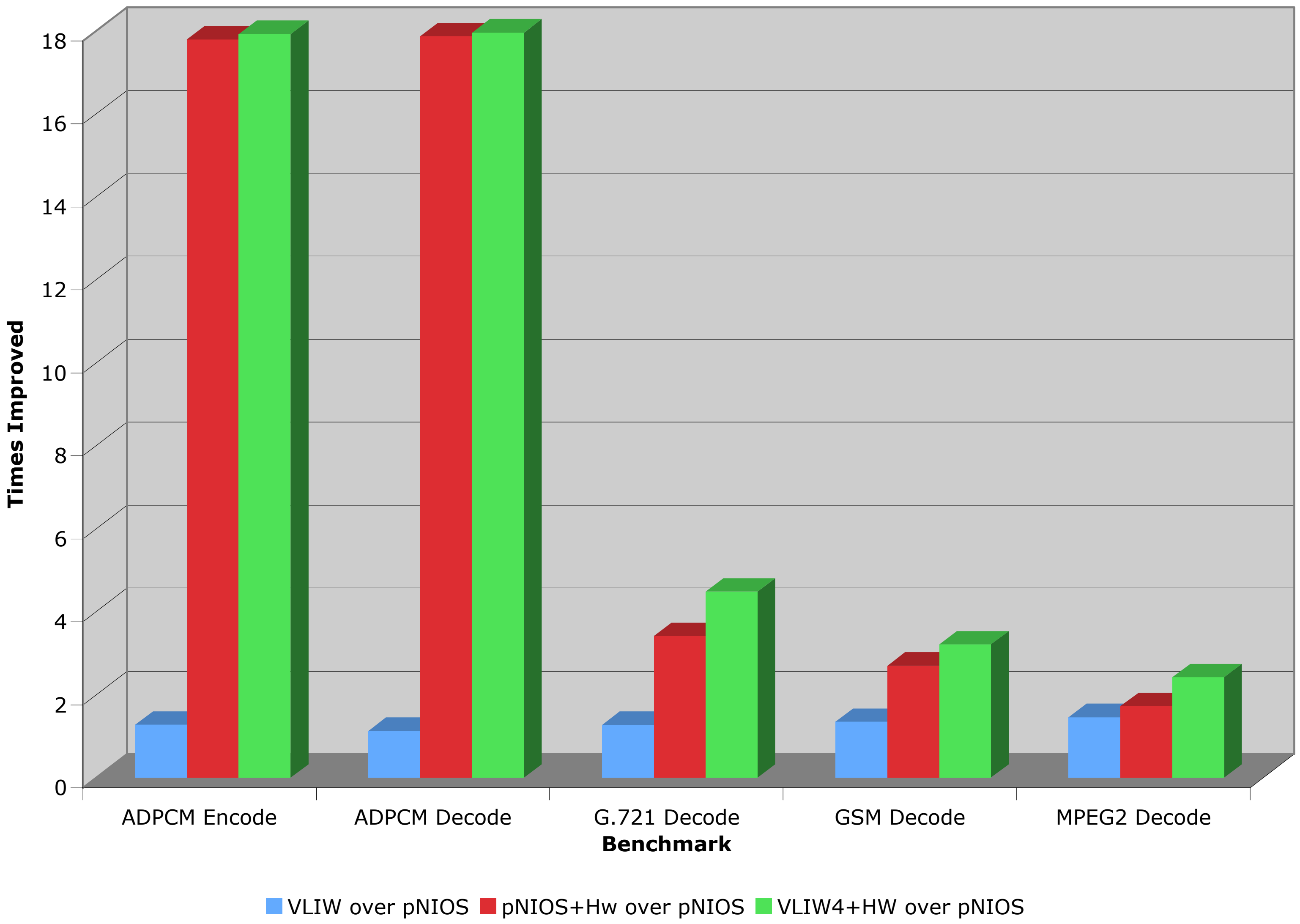
Kernel Acceleration: 7x to 345x Software vs. HW-Accelerated



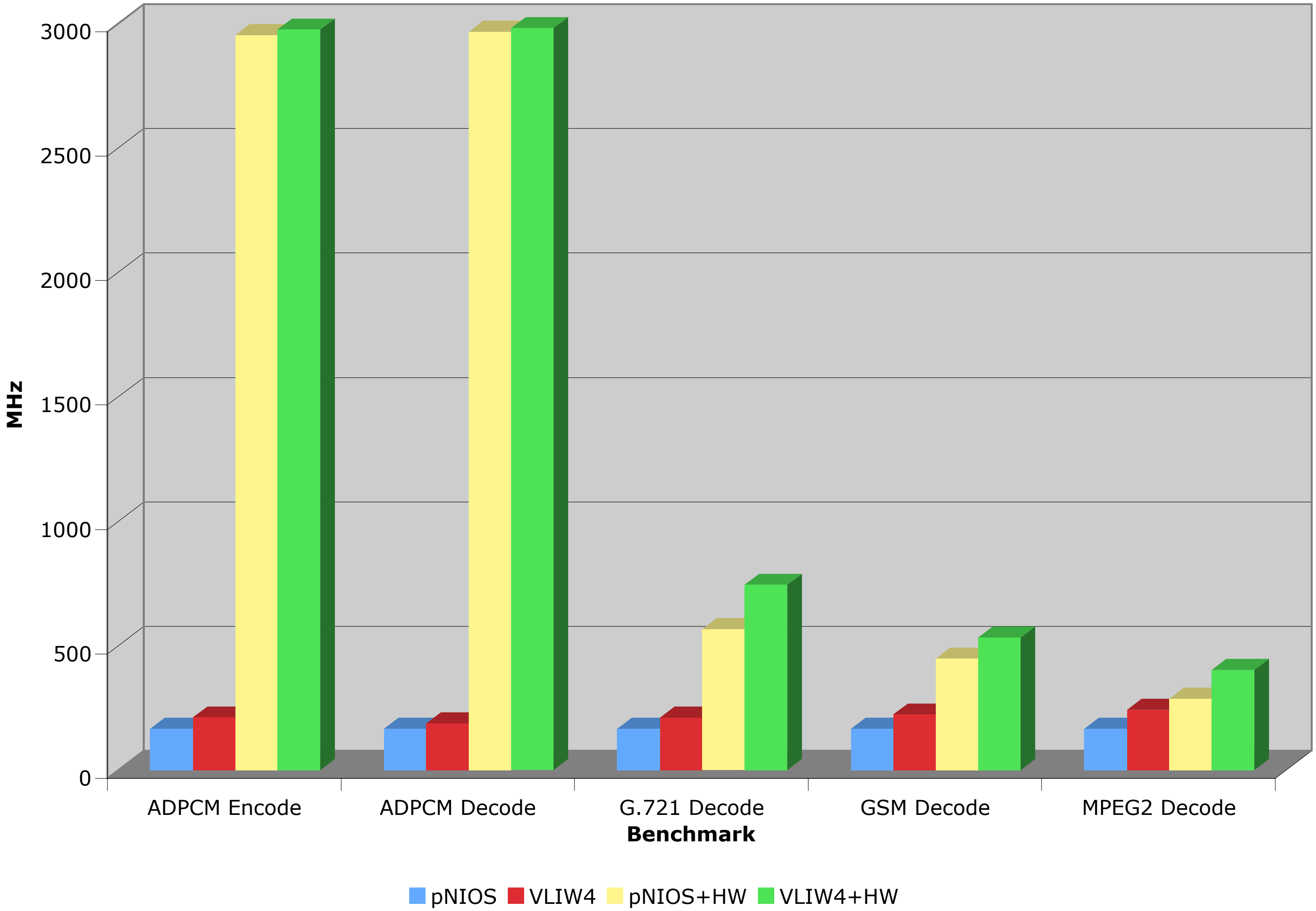
Application Acceleration: 1.8x to 154x Software vs. HW-Accelerated



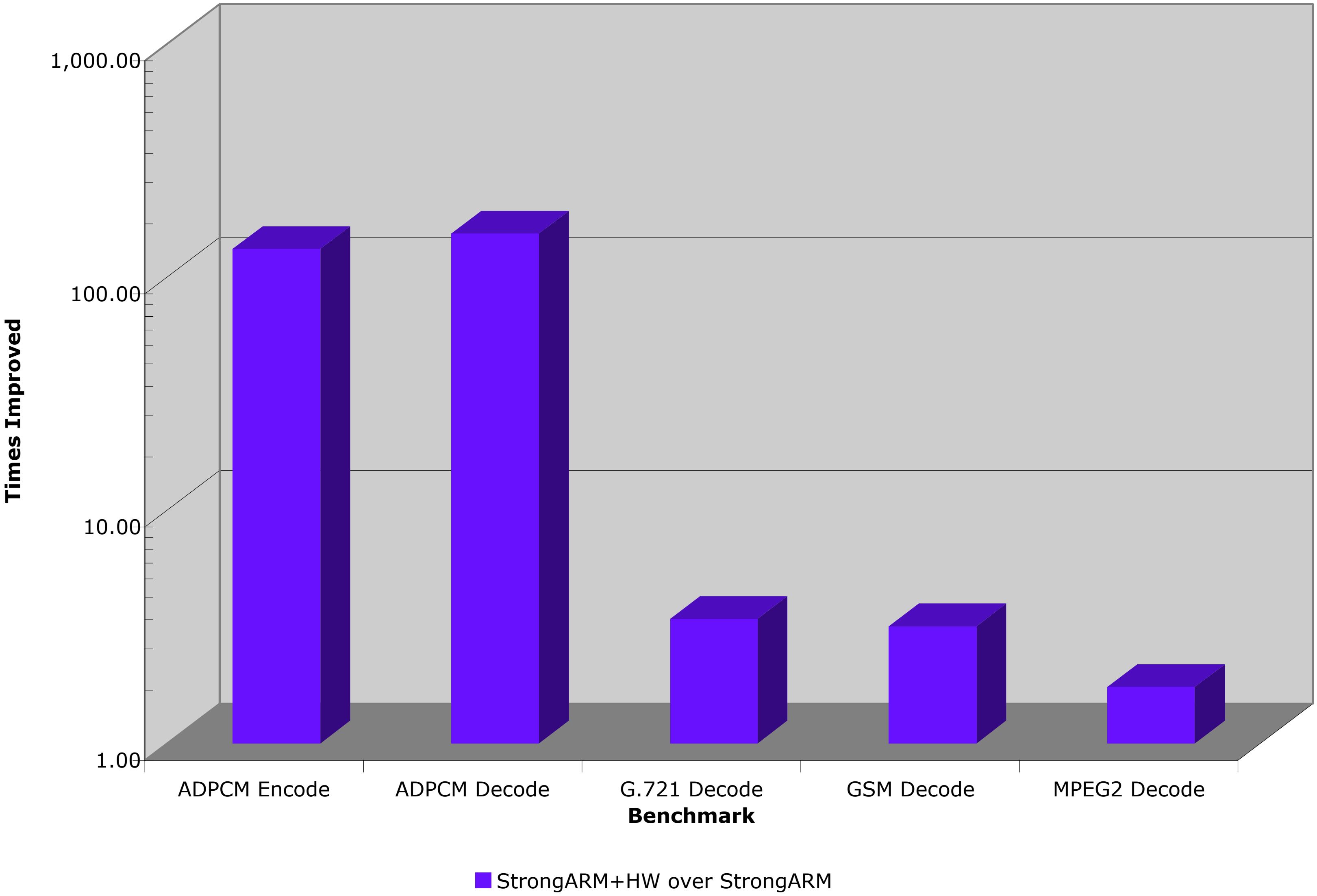
Application Acceleration: pNIOS



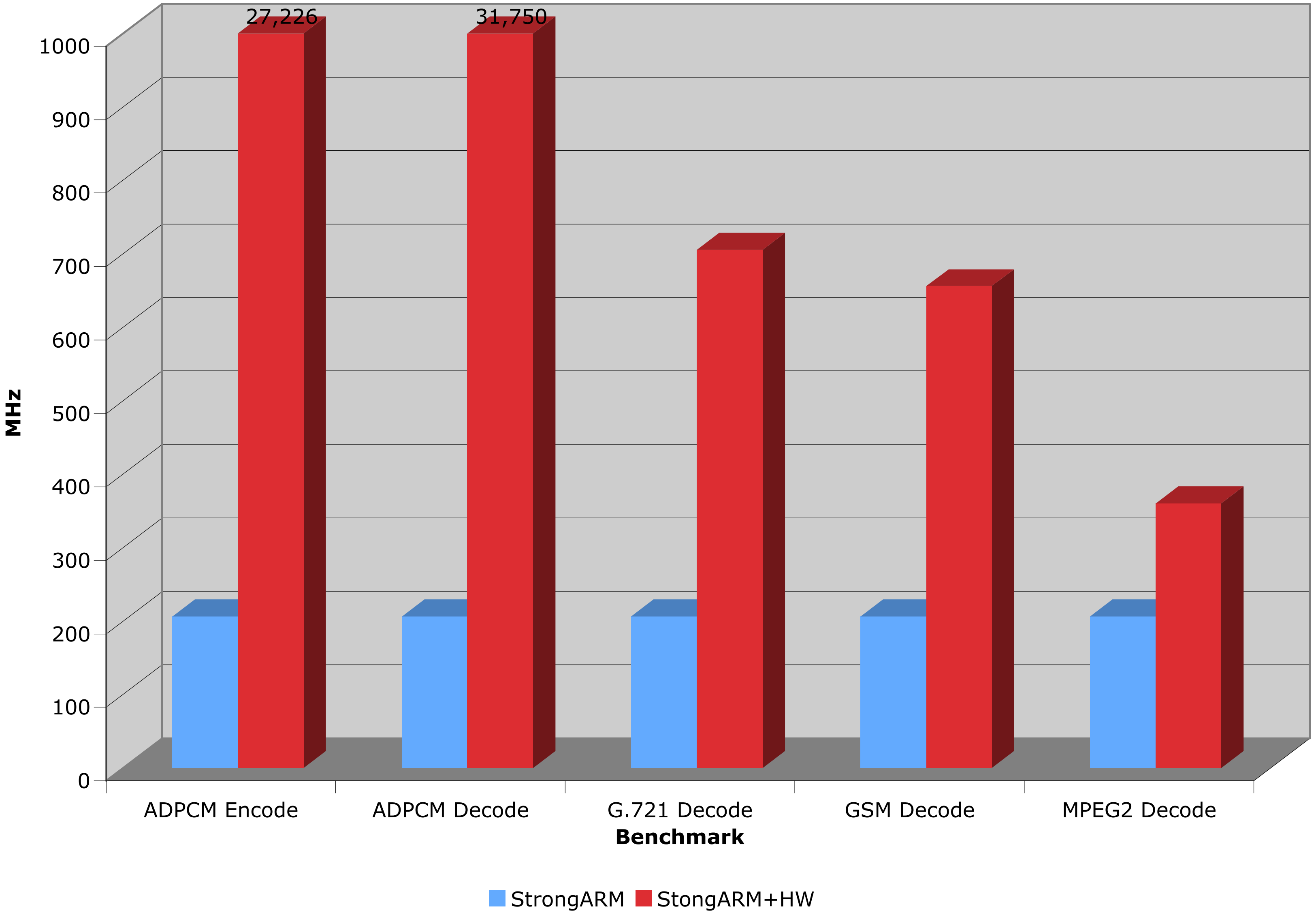
Effective MHz for pNIOS



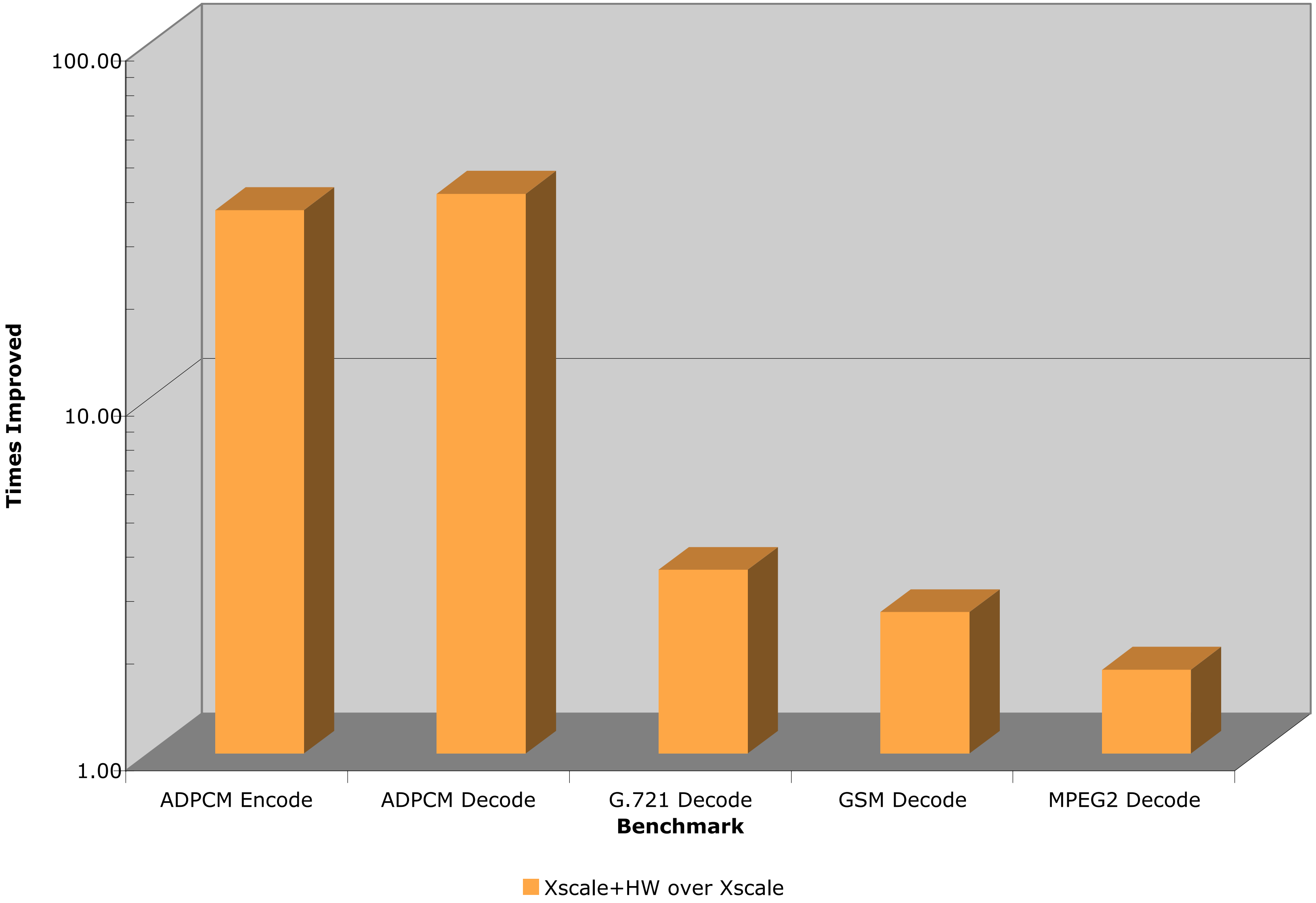
Application Acceleration: StrongARM



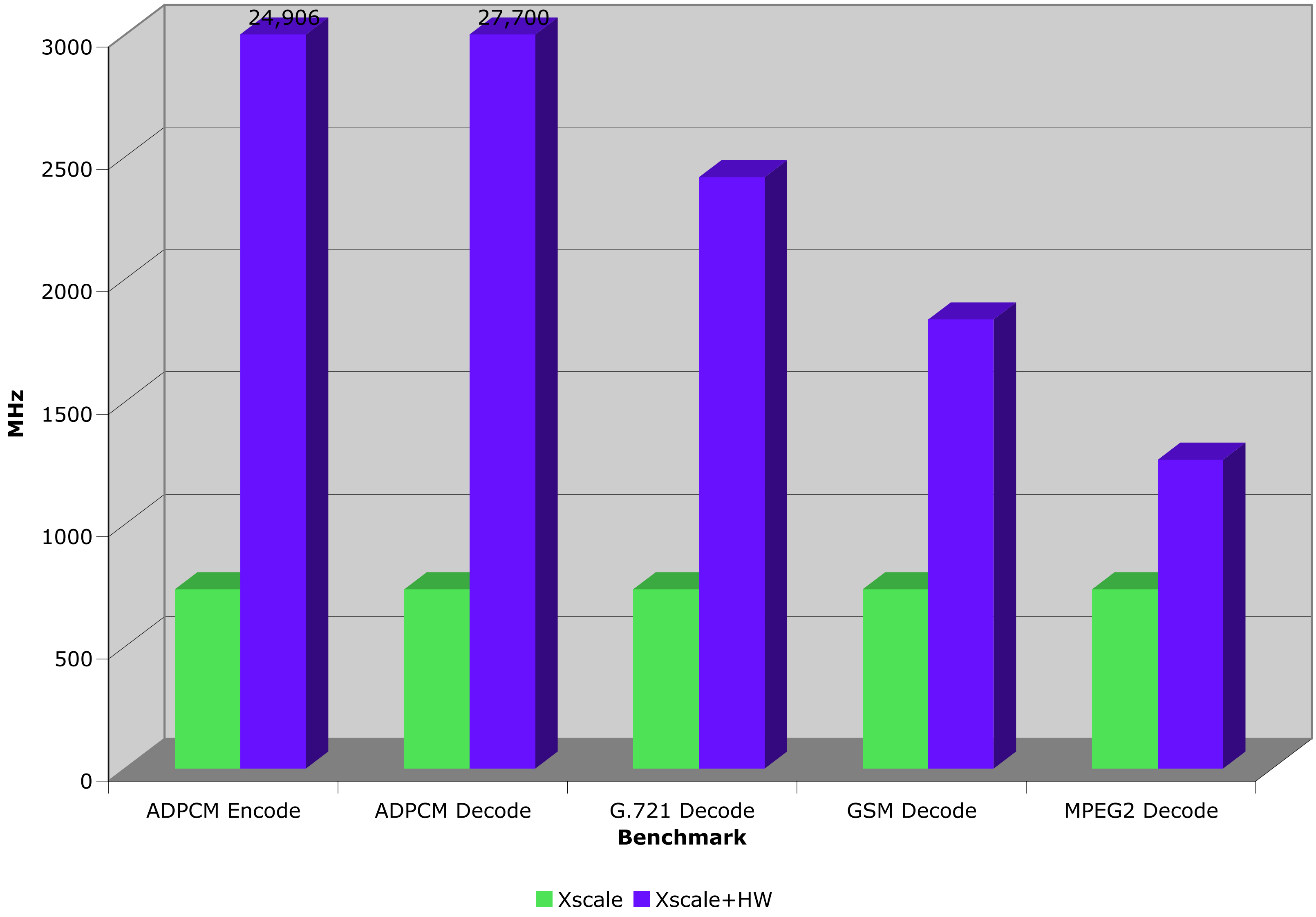
Effective MHz for StrongARM



Application Acceleration: XScale

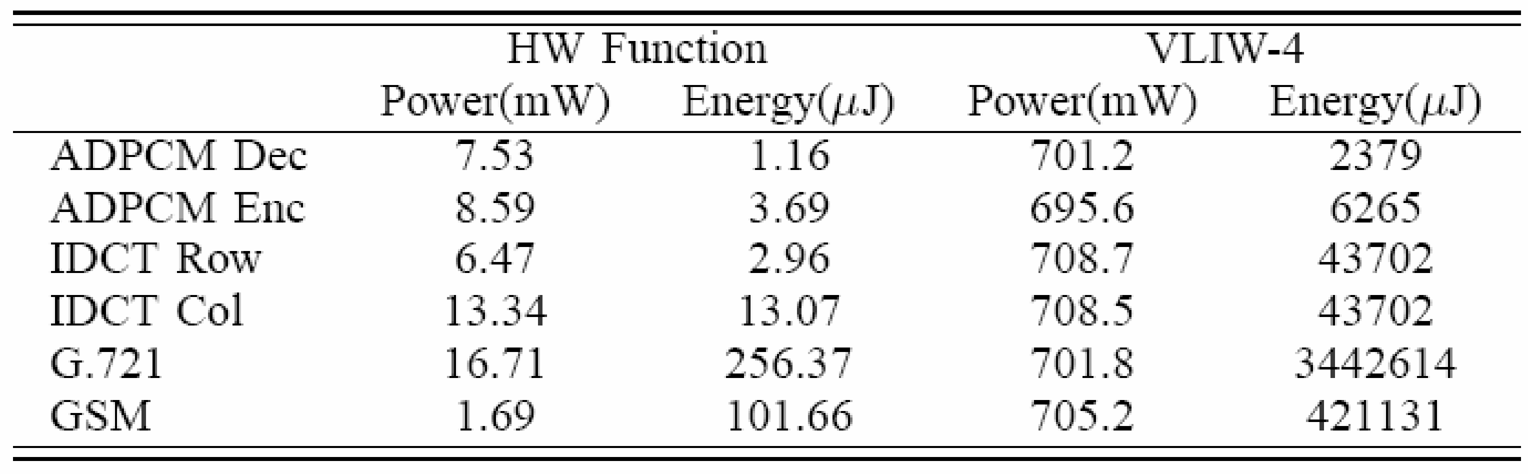


Effective MHz for XScale



Power/Energy Savings

- Software on 4 wide VLIW in 160nm ASIC vs. Hardware Functions in 160nm ASIC
- Power Reduction: 42x to 418x
- Energy Reduction: 1,700x to 60,000x



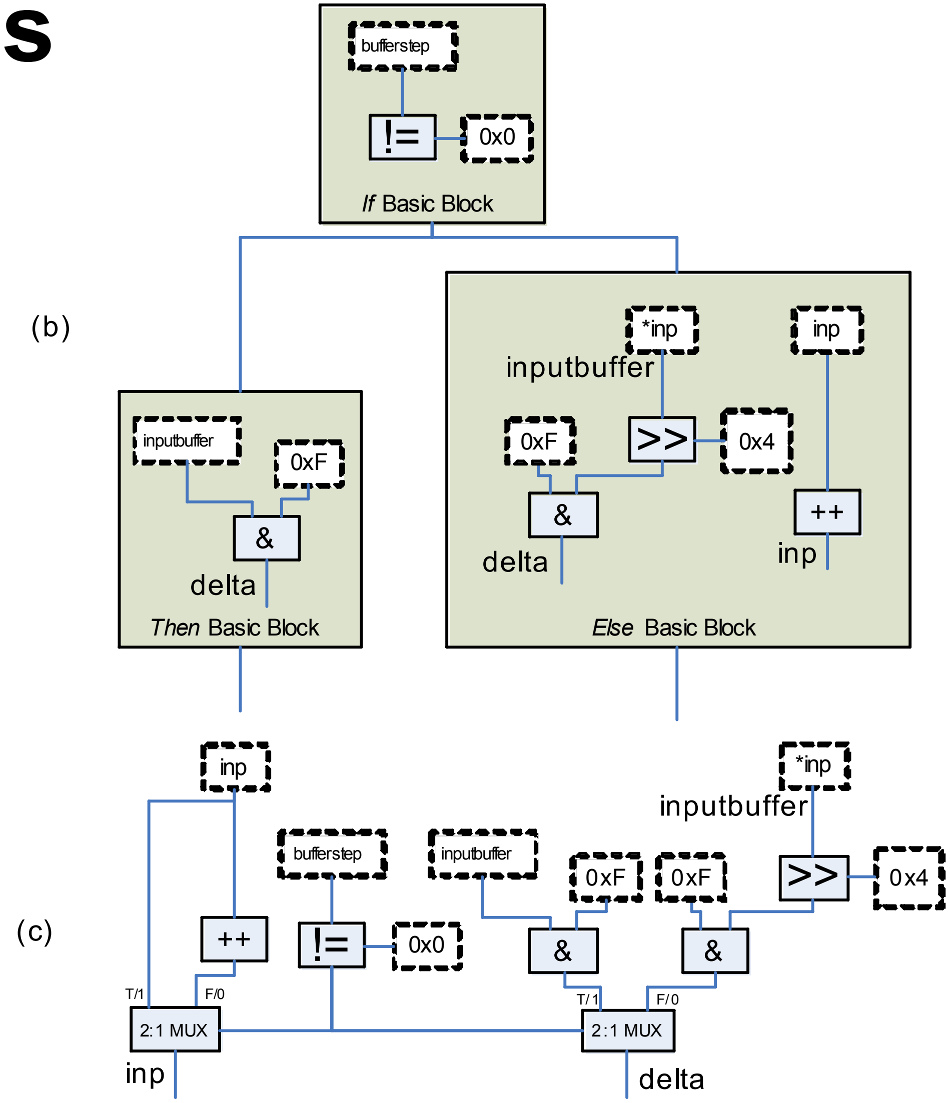
	HW Function		VLIW-4	
	Power(mW)	Energy(μJ)	Power(mW)	Energy(μJ)
ADPCM Dec	7.53	1.16	701.2	2379
ADPCM Enc	8.59	3.69	695.6	6265
IDCT Row	6.47	2.96	708.7	43702
IDCT Col	13.34	13.07	708.5	43702
G.721	16.71	256.37	701.8	3442614
GSM	1.69	101.66	705.2	421131

Automatic Conversion of Software Functions into Hardware Functions

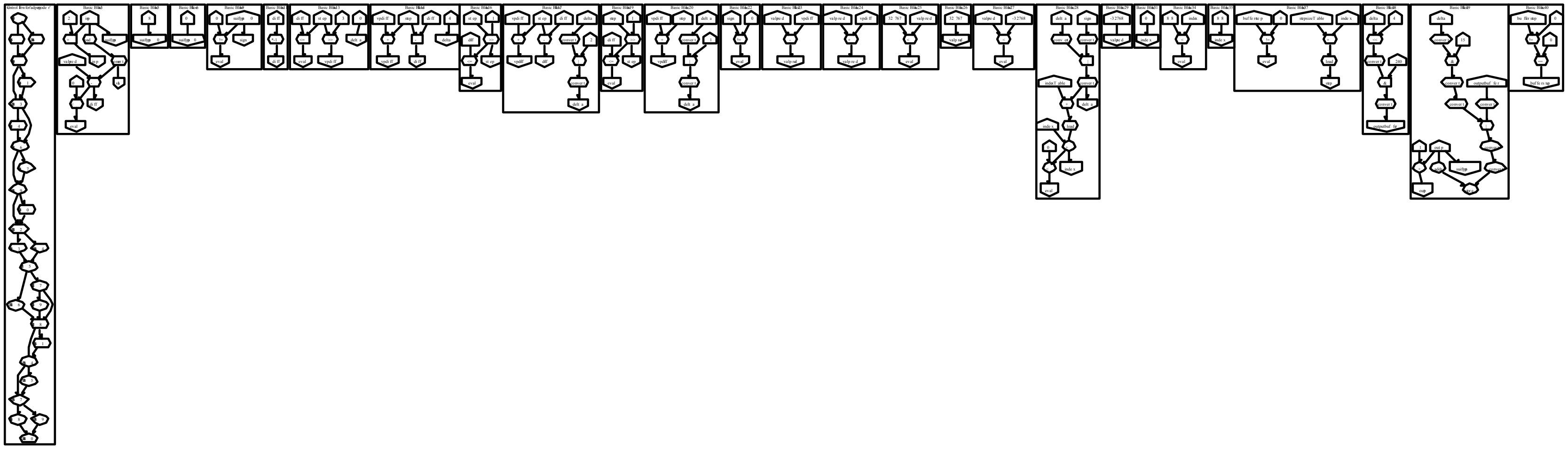
- Specified hardware functions are converted automatically into hardware
- Part of the “compiler”
- Prototype compiler is working in laboratory

(a)

```
If (bufferstep) {
  delta = inputbuffer & 0xf;
} else {
  inputbuffer = *inp++
  delta = (inputbuffer >> 4) & 0xf;
}
```

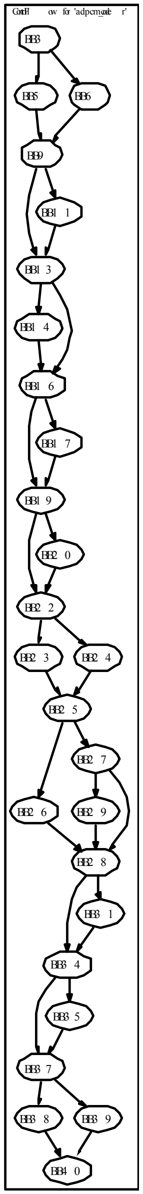


ADPCM - CDFG



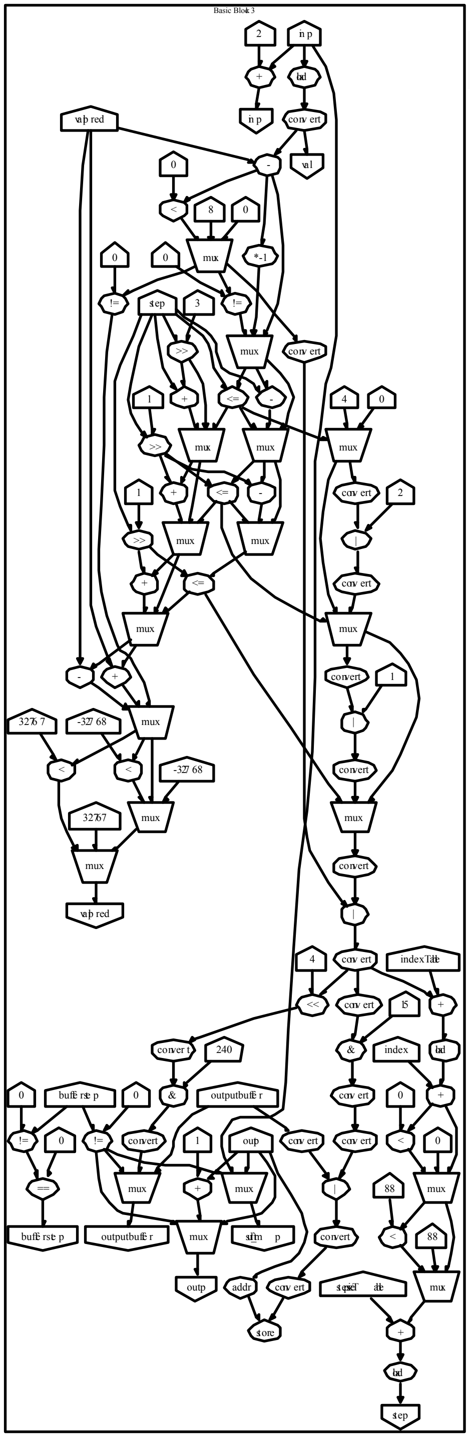
ADPCM - CFG

- Control flow dominated
  - □ *if-then-else*
- ILP < 2
- Branch prediction is poor
- Poor inherent parallelism
- DFGs contain variable length critical paths
  - □ Cycle time for RTL conforms to longest DFG



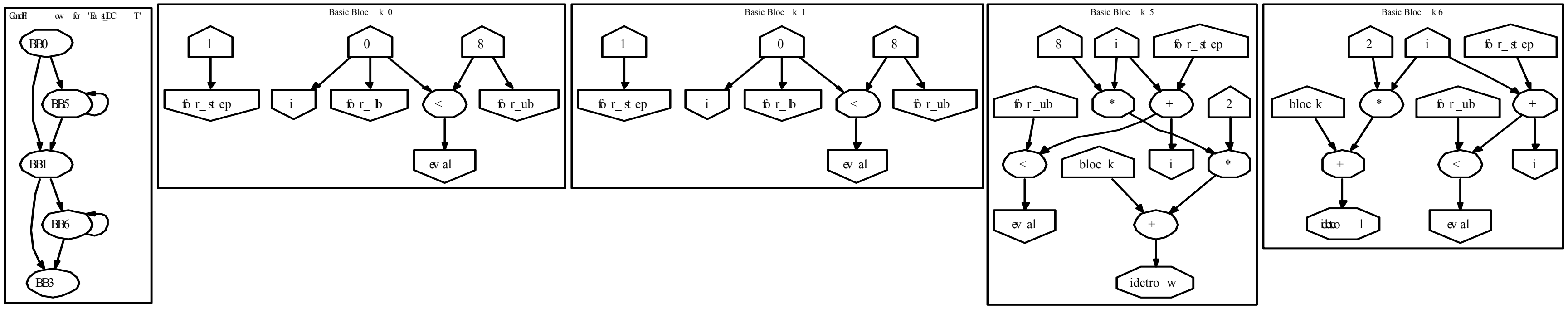
ADPCM - SDFG

- Single SDFG
- Removal of clock
  - Combinational/Data Flow
  - Cycle compression
- Significant increase in parallelism

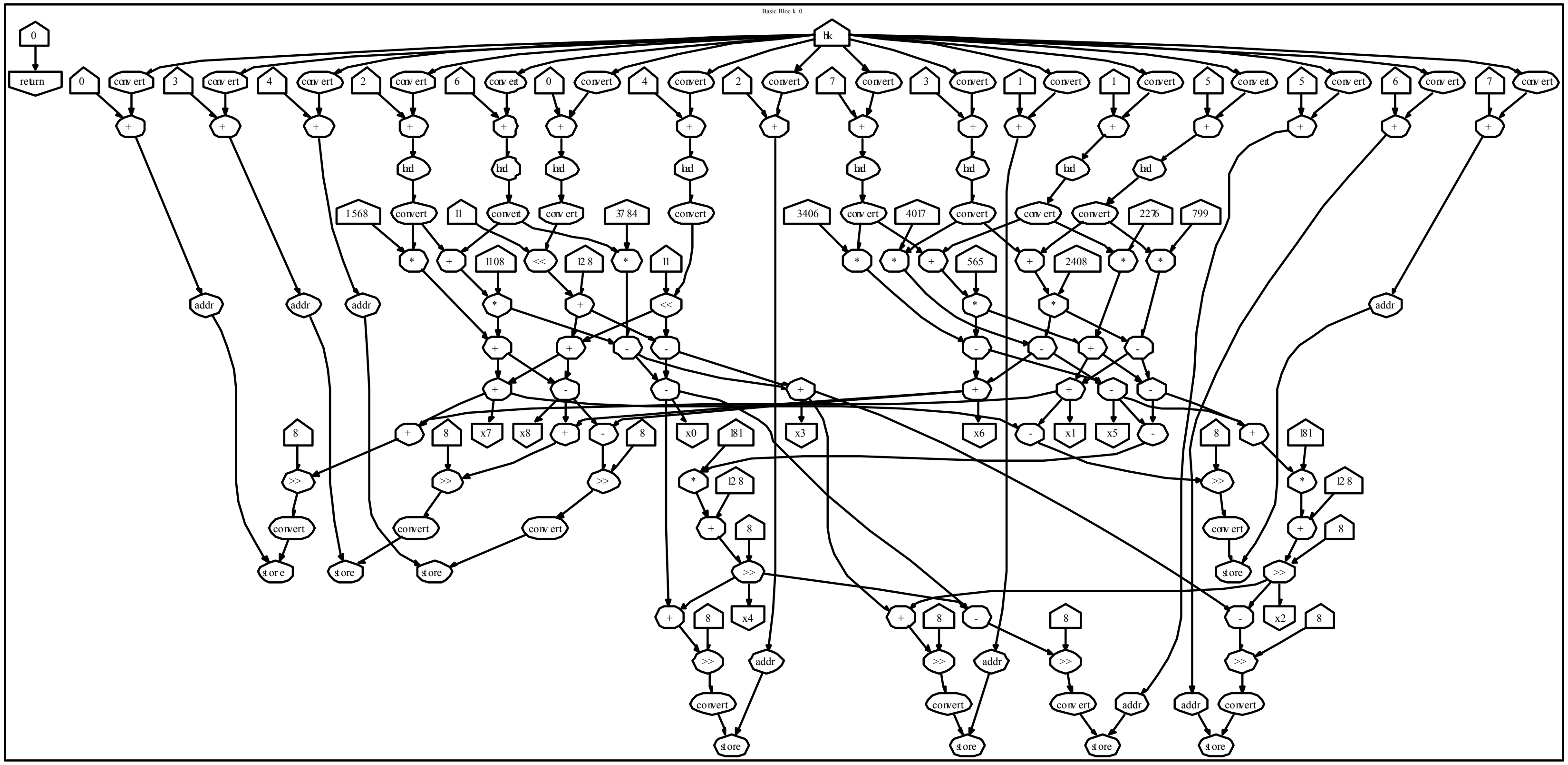


MPEG II (IDCT) - CDFG

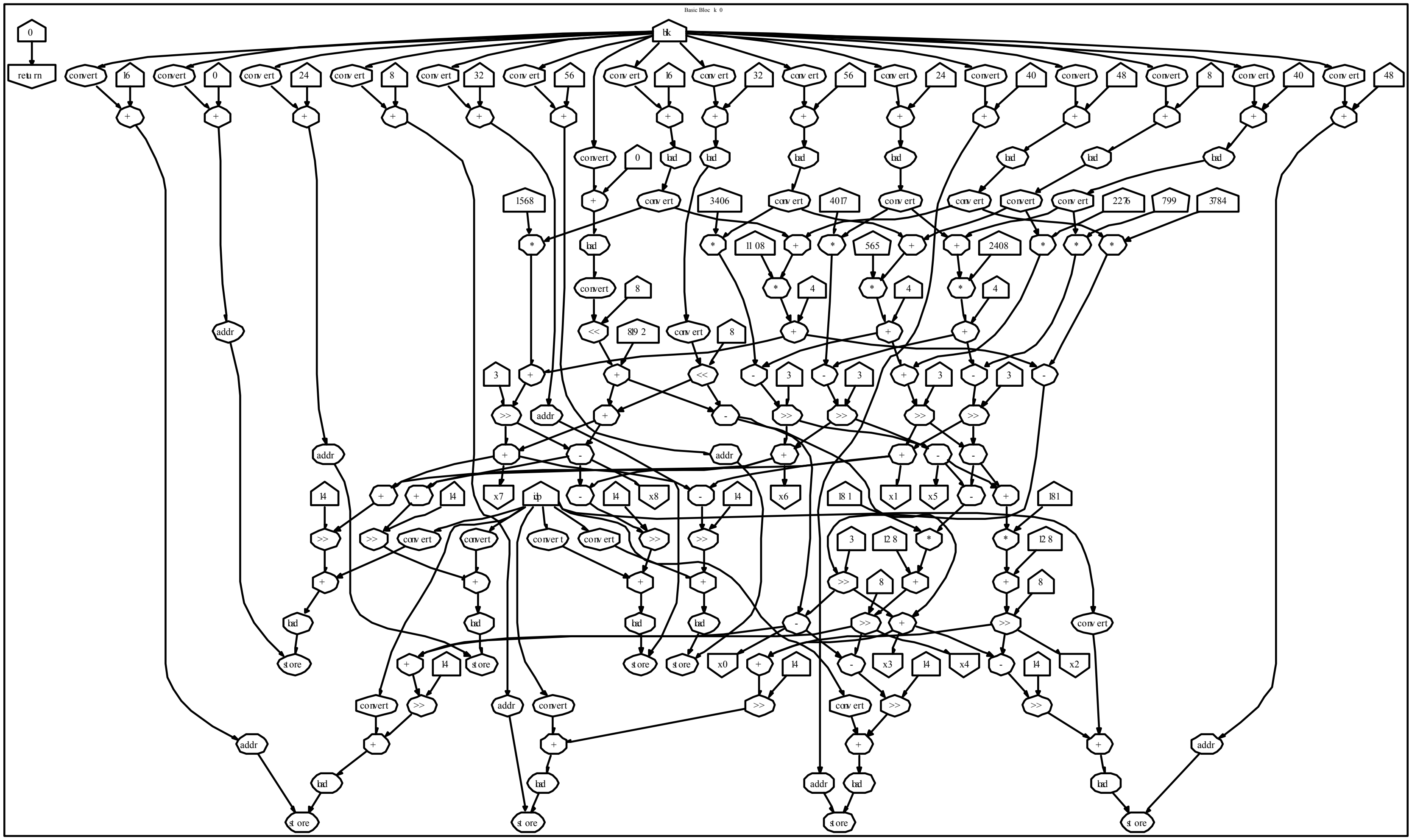
- Two nested loops within fast_IDCT function
  - idct_row and idct_col
- In this example, control flow is segregated from most of the work



MPEG II (IDCT) - CDFG (row)



MPEG II (IDCT) - CDFG (col)



A VLIW Processor with Hardware Functions: Increasing Performance While Reducing Power

- **For FPGA, Structured ASIC and ASIC technologies**
- VLIW Architecture
  - □ Up to 4x performance improvement for software alone
  - □ Full 32-bit Processor
    - Real applications
    - Altera NIOS II Instruction Set Architecture
- "Hardware Functions"
  - □ Convert critical software kernels into hardware
  - □ **9x to 332x Performance improvement**
  - □ **42x to 418x Power reduction**
- **Productivity** through Design Automation
  - □ Source: Unmodified C programs
  - □ **"Compile" software into hardware functions**
- **Contact**
  - □ **Prof. Ray Hoare hoare@ece.pitt.edu**
  - □ **Prof. Alex Jones akjones@ece.pitt.edu**