

Evolution of DSP Processors

Kartik Kariya

EE, IIT Bombay

Agenda

- **Expected features of DSPs**
- **Brief overview of early DSPs**
- **Multi-issue DSPs**
- **Case Study: VLIW based Processor (SPXK5) for Mobile Applications**

Why DSPs?

Most DSP tasks require:

- **Real- time processing**
- **Repetitive numeric computations**
- **Attention to numeric fidelity**

Processors must perform these tasks efficiently while minimizing:

- **Cost**
- **Power**
- **Memory use**
- **Development time**

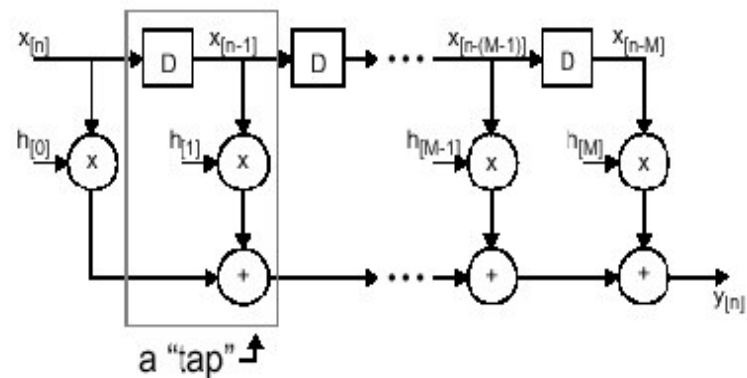
Features of DSPs

- **DSPs architecture is driven by algorithms**
- **Algorithms puts requirement for exhaustive computations**
eg. Consider FIR filter example

$$Y(n) = \sum_{K=0}^{M-1} h(n) x(n-k)$$

Each tap (M taps total) requires:

- **Two data fetches**
- **Multiply**
- **Accumulate**



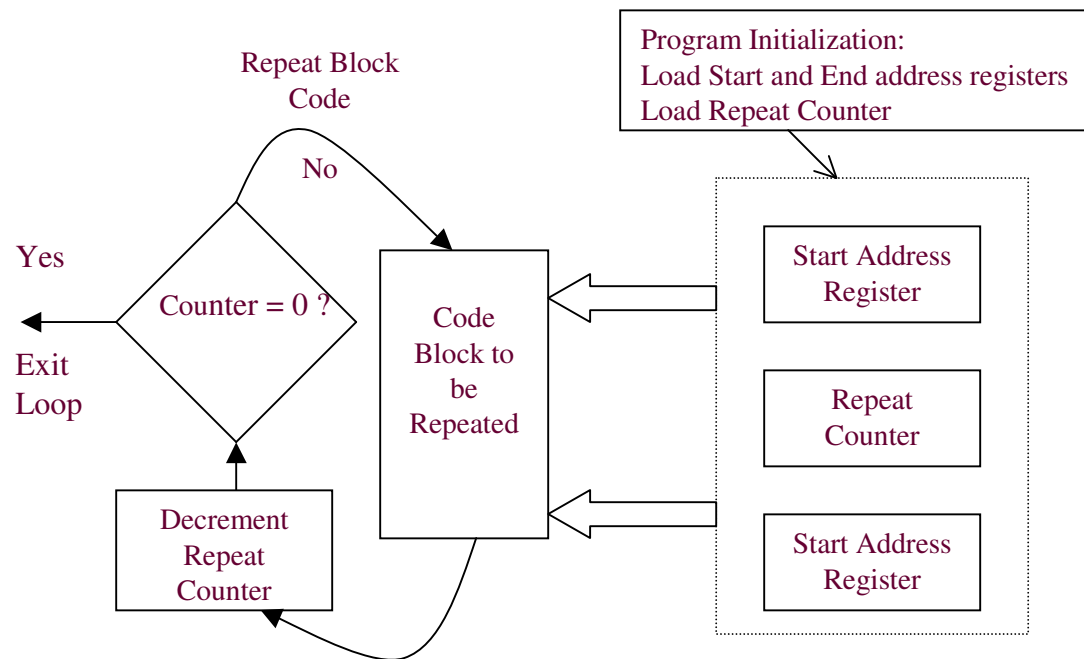
Features of DSPs

Features Common to Most DSP Processors

- **Specialized hardware performs all key arithmetic operations in 1 cycle.**
- **Specialized addressing modes for efficient Memory access**
 - e. g.
 - Auto increment
 - Circular addressing
 - Bit- reversed (for FFT)
- **Hardware support for managing numeric fidelity**
 - Guard bits
 - Saturation

Features of DSPs Conti..

- **Zero-Overhead Looping**
 - specialized hardware for “test and branch”
 - loop nesting



Features of DSPs Conti..

- **Specialized, complex instructions**
 - To make maximum advantage of processor hardware
 - Minimize the memory space required for storing data
- **Multiple operations per instruction**
- **I/O handling mechanism with no intervention to computational units**
- **Other features like on chip ADC,DAC, DMA controller etc.**

Brief Overview Of Early DSPs

- **First Generation DSPs :**
 - e.g Texas Instruments TMS32010
 - 16 Bit Multiplier, and 32 bit Accumulator
 - Issue and execute one instruction per clock cycle
 - Performance 6-8 MHz (390 ns - MAC instruction)
- **Second Generation DSPs :**
 - e.g ADSP-21xx, TMS320C2xx, DSP560xx
 - Pipelined to some extent
 - 20-50Mhz (75 ns MAC)
 - 16/24/32bit instructions
 - Makes Instruction set complicated, irregular
- **Typically used in consumer and telecom products that have modest DSP performance requirements**

Brief Overview Of Early DSPs contd...

- **Mid Range/Third Generation DSPs :**
 - Higher clock speeds – 100-150Mhz
 - Parallel execution units– multipliers, adders
 - Deeper pipelines, parallelism
 - Wider data buses, wider instruction words
 - Increase in cost and power consumption offset by performance
 - 20ns MAC
- Used in higher performance DSP tasks
- Wireless Telecom, high-speed modems

Examples : TMS320C54x, DSP16xxx (Lucent) etc.

Problems : Difficult assembly, compiling

Multi-issue DSPs

- **Goals**
 - **High Performance**
 - **Compiler friendly architecture**
- **Simple instructions, 1 operation/instruction**
- **Issue/execute instructions in parallel groups**
- **3ns MAC throughput**
- **Targeted at demanding computational requirements**

Two classes of multi-issue

- **VLIW**
- **Superscalar**

Multi-issue DSPs Contd...

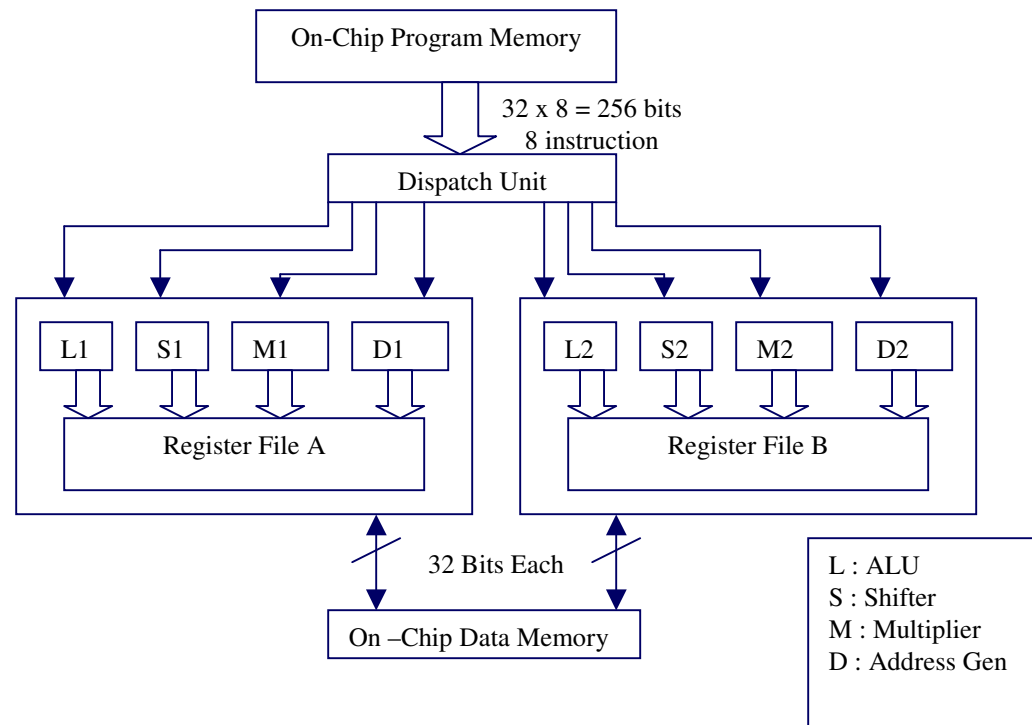
- **Very Long Instruction Word (VLIW) Class**
 - TMS320C62xx, first multi-issue (VLIW), introduced in 1996
 - Large number of parallel execution unit
 - Typically issue 4-8 instructions / cycle (VLIW)
 - Assembly programmer / compiler decide parallel instruction grouping depending on data dependencies and resource contention.
 - Instruction groups do not change in execution
 - Large number of instruction decoders, buses, registers and hence memory bandwidth

Problems : High-Energy consumption

Usage : high computational applications e.g. Cellular base station

Multi-issue DSPs Cont...

- **TMS320C6xx Execution Unit**



Multi-issue DSPs Cont...

- **Superscalar Class**
 - Special hardware decides parallel instruction grouping considering data dependency, resource contention
 - Instruction groups can change in execution depending on data access, loop execution etc.

Problems :

- Difficult to predict execution times hence not suitable for real time applications
- High energy, memory usage

Single Instruction Multiple Data Technique

Single Instruction Multiple Data technique (SIMD)

- Execute multiple instances of the same operation in parallel using different data.
- Combined with VLIW / Superscalar / Conventional
- Boosts performance in vector heavy operations such as multimedia applications.
- Based on added parallel execution units(e.g. ADSP-2116x) or logical split of existing execution units(e.g TigerSharc)

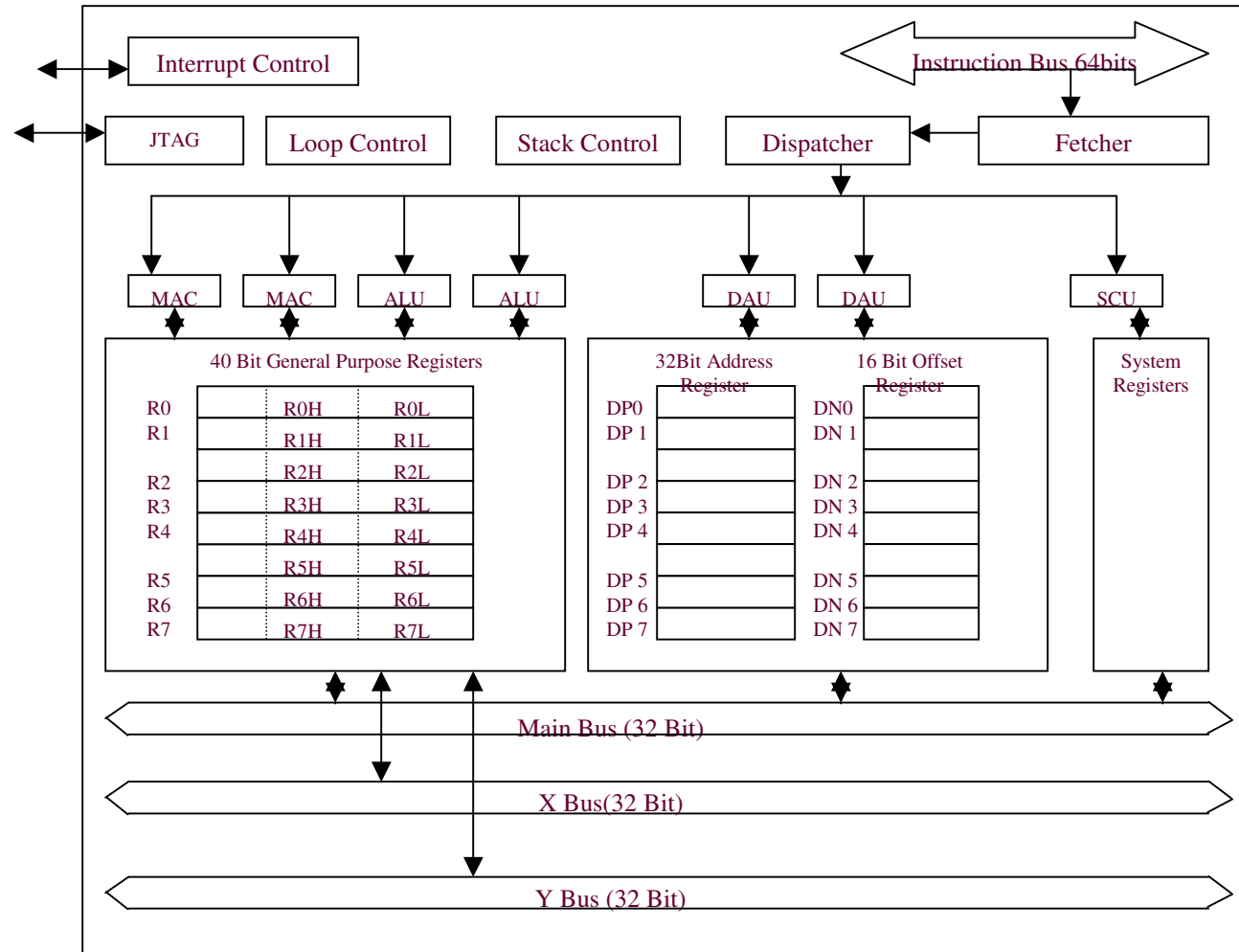
Problems :

- Must arrange data in memory
- Algorithm re-organization to use processor resources
- Not effective for algos that are inherently serial

Case Study: VLIW based Processor (SPXK5) for Mobile Applications

- **Requirements**
 - Higher Processing Power for multimedia applications like video codecs, Speech codecs, speech recognition systems etc executing simultaneously
 - Fast beat rate.
 - Minimum Power consumption.
- **Architectural overview**
 - In incorporates customized VLIW approach as well as SIMD features to give better performance.
 - The functional units consist of Two multiply-accumulate (MAC), two arithmetic units (ALU), two data address units (DAU) for load and store and System control unit (SCU) for branch, zero overhead looping, and conditional execution.

Case Study: VLIW based Processor (SPXK5) for Mobile Applications contd.



Case Study: VLIW based Processor (SPXK5) for Mobile Applications contd.

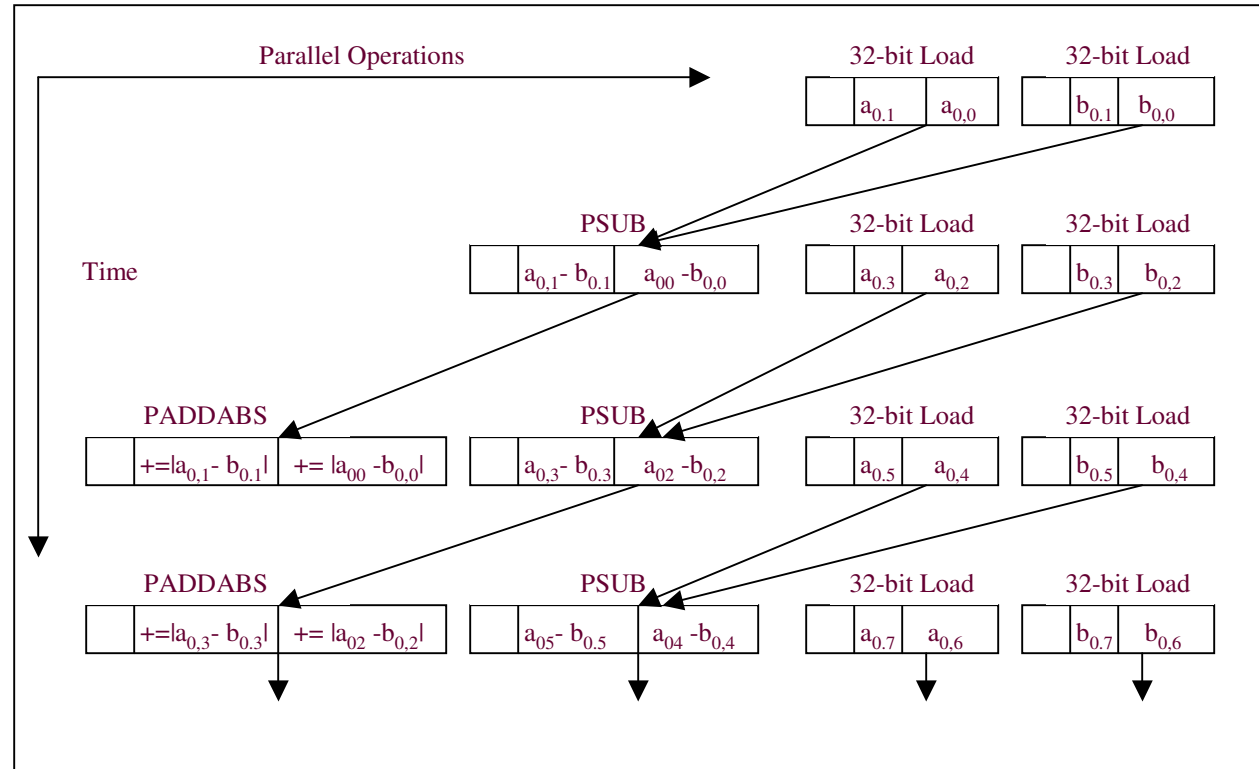
Features

- **Operational Frequency 250 MHz; Avg. power Consumption 0.15 mW/MIPS at 1.5 V**
- **Maximum four functional units work simultaneously.**
- **16 Kbyte instruction cache. Six-stage pipeline: Instruction fetch, dispatch queue, decode, DP register update, Execution phase I and II.**
- **Instruction: 16 or 32 bits long; Instruction packet size 16 to 64 bits. gives higher code density.**
- **Eight special SIMD instructions (PADD, PSUB, PSHIFT, PADDABS, PACKV etc) to take advantage of data-level parallelism.**
- **SIMD Instructions useful to implement DSP algorithms such as video encoding/decoding,FFT etc.**

Case Study: VLIW based Processor (SPXK5) for Mobile Applications contd.

e.g. Implementation of mean absolute error (MAE) required in Motion Estimation for video codec

$$MAE = \sum_{M=0}^{M-1} \sum_{N=0}^{N-1} |A_{mn} - B_{mn}|$$



Conclusion

- **DSP Processor performance has increased substantially over the years**
- **Drivers for evolution of DSPs: speed, energy, memory usage, cost**
- **Focus is on compiler-friendly architectures**
- **DSP processor architectures is increasingly being specialized for specific applications.**