Multimedia Systems

Lecture 01 Introduction

Saeid Nooshabadi
APPLICATIONS OF VLSI SIGNAL PROCESSING

• Embedded digital signal processing is everywhere!

• Examples:
  – Speech
  – Audio
  – Video
  – Radio/wireless
  – Any application that processes signals in the digital domain
TYPICAL ALGORITHMS

• Filtering: FIR, IIR, with fixed coefficients or adaptive
• Encoding/decoding
• Compression/decompression
• Frequency-domain processing
• Etc.
TYPICAL NUMBERS

• Speech: 8 kHz, 12-16 bits
• Audio: 44 kHz, 16-24 bits, two channels (stereo)
• Video, various formats, e.g.:
  – HDTV approx. 2000 by 1000 pixels at 50 frames per second resulting in data rates of 100 MHz, 3 colors of 8-12 bits each
STREAMING VS. BLOCK-BASED

• **Streaming** data:
  – Data samples are processed as they arrive
  – Requires little local storage
  – Time-domain processing

• **Block-based** processing:
  – Stores incoming data until some block size is filled
  – Processes entire block
  – Think e.g. of an FFT (Fast Fourier Transform) or DCT (Discrete Cosine Transform)
IMPLEMENTATION PLATFORMS

- General-purpose processor, such as a Pentium
- Digital signal processor:
  - Much better suited (parallel arithmetic in data path, support for “multiply-accumulate” operation, Harvard architecture for parallel access to data and program memory, etc.)
- Very large instruction word (VLIW) processor:
  - Many parallel arithmetic units in data path, each controlled by appropriate bits in instruction word
- Processor arrays
- User-defined architectures
- Dedicated logic (implemented by standard cells or on FPGA)
TEXAS INSTRUMENTS DSP
TMS320C-6455 (1 GHz)
VLIW ARCHITECTURE
(e.g. PHILIPS/NXP TRIMEDIA)

- long instruction words e.g. \((3\times7+4)\times25=625\) bits
- many ports on the register file e.g. 75
MAPPING PROBLEM

- How do we get the most efficient implementations of DSP algorithms on our platforms?

- Optimization criteria:
  - Fastest
  - Smallest
  - Minimal energy
  - Shortest design time

- In general, *flexibility* comes at the expense of *efficiency*. 
EXAMPLE: SITEL’S SC14480 SINGLECHIP DECT PROCESSOR

VLSI Signal Processing
VLSI DSP Issues

• Non-Terminating Programs Require Real-Time Operations

• Applications dictate different speed constraints
  – (e.g., voice, audio, cable modem, settop box, Gigabit ethernet, 3-D Graphics)

• Need to design Families of Architectures for specified algorithm complexity and speed constraints

• Methods of representations of DSP Algorithms
Typical DSP Programs

• Usually highly real-time, design hardware and/or software to meet the application speed constraint

• Non-terminating Examples

\[
\text{for } n = 1 \text{ to } \infty \\
y(n) = a \cdot x(n) + b \cdot x(n - 1) + c \cdot x(n - 2) \\
\text{end}
\]
Area-Speed-Power Tradeoffs

• 3-Dimensional Optimization (Area, Speed, Power)
• Achieve Required Speed, Area-Power Tradeoffs
• Power Consumption ($P = C \times V^2 \times f$)
• Latency reduction Techniques ➔ Increase in speed or power reduction through lower supply voltage operation
• Since the capacitance of the multiplier is usually dominant reduction of the number of multiplications is important (this is possible through strength reduction)
Example: $y(n) = a \cdot x(n) + b \cdot x(n-1) + c \cdot x(n-2)$

- Graphical Representation Method 1: Block Diagram
  - Consists of functional blocks connected with directed edges, which represent data flow from its input block to its output block
Graphical Representation Method 2: Signal-Flow Graph

- SFG: a collection of nodes and directed edges
- Nodes: represent computations and/or task, sum all incoming signals
- Directed edge \((j, k)\): denotes a linear transformation from the output signal at node \(j\) to the input signal at node \(k\)
- Linear SFGs can be transformed into different forms without changing the system functions. For example, Flow graph reversal or transposition is one of these transformations (Note: only applicable to single-input-single-output systems)
- Usually used for linear time-invariant DSP systems representation
Graphical Representation Method 3: Data-Flow Graph

- DFG: nodes represent computations (or functions or subtasks), while the directed edges represent data paths (data communications between nodes), each edge has a nonnegative number of delays.
- DFG captures the data-driven property of DSP algorithm: any node can perform its computation whenever all its input data are available.
- Each edge describes a precedence constraint between two nodes in DFG:
  - Intra-iteration precedence constraint: if the edge has zero delays
  - Inter-iteration precedence constraint: if the edge has one or more delays
  - DFGs and Block Diagrams can be used to describe both linear single-rate and nonlinear multi-rate DSP systems
  - Fine-Grain DFG
Examples of DFG

- Nodes are complex blocks (in Coarse-Grain DFGs)
- Nodes can describe expanders/decimators in Multi-Rate DFGs
Gourse Grading

• Homework (10)
• Project (25)
• Mid Sem (20)
• Final (45)
DE2-70 Development Board

The DE2-70 board has many features that allow the user to implement a wide range of designed circuits, from simple circuits to various multimedia projects.
DE2-70 Features (#1/2)

- The Altera DE2-70 development and education board, equipped with almost 70,000 LEs of Altera Cyclone® II 2C70,
- 2-Mbyte SSRAM
- Two 32-Mbyte SDRAM
- 8-Mbyte Flash memory
- SD Card socket
- 4 pushbutton switches
- 18 toggle switches
- 18 red user LEDs
- 9 green user LEDs
- 50-Mhz and 28.63-Mhz oscillators for clock sources
- 24-bit CD-quality audio CODEC with line-in, line-out, and microphone-in jacks
- VGA DAC (10-bit high-speed triple DACs) with VGA-out connector
- 2 TV Decoder (NTSC/PAL) and TV-in connector
DE2-70 Features (#2/2)

- 10/100 Ethernet Controller with a connector
- USB Host/Slave Controller with USB type A and type B connectors
- RS-232 transceiver and 9-pin connector
- PS/2 mouse/keyboard connector
- IrDA transceiver
- 1 SMA connector

- Two 40-pin Expansion Headers with diode protection
- Size: 153*203 mm

DE2-70 Layout
SOPC Builder Tool
Topics

• Topics include; introduction to custom digital processors including DSP hardware, high-speed digital design techniques, modern chip design methodologies, hardware and software co-design, advanced programming paradigms including state machines and concurrent processes, real-time programming and operating systems.
References

- Additionally, you may find the following reference books helpful:
  - Embedded Systems Architecture: A Comprehensive Guide for Engineers and Programmers (Embedded Technology) by Tammy Noergaard
Conclusion

• Digital architectures are used for DSP applications
  – Speech
  – Audio
  – Video
  – Radio/wireless
  – Any application that processes signals in the digital domain