Hardware Software Co-design and SoC

Neeraj Goel
IIT Delhi
Introduction

- What is hardware software co-design
  - Some part of application in hardware and some part in software
  - Mpeg2 decoder example
Introduction

- **MPEG2 Decoder example**
  - Prediction and IDCT are most computation intensive
    - Best candidates for hardware
MPEG2 Decoder example

- Deadline time for each frame = 1/25 sec = 40msec
- Achievable with 1GHz processor in 20msec
- With 100MHz processor = 200msec
- IDCT and prediction is 90% computation
  - A hardware IDCT and prediction = 10msec
- Suggested architecture
  - 10% computation on hardware + 90% on hardware
  - 20msec + 10msec + overhead < 40msec
Why Hardware Software Codesign

- **Hardware Advantages**
  - Less area, low power, high performance
  - Low flexibility (difficult to modify)
  - More design time

- **Software Advantage**
  - More area, power, less performance
  - High flexibility (easy to modify)
  - Less design time

- **Embedded system issues**
  - Area, power and performance matters
  - Solution is hardware-software codesign
Issues with Hardware-Software Codesign

- Partitioning
  - Selecting modules for hardware and software
- Testing
  - Co-simulation and verification
- Code generation
  - Software controls for hardware
- Design issues
  - Hardware Software interface
    - Communication between h/w and s/w
Target Architecture

- Platform based solutions (fixed interface)
  - Use standard communication buses (AMBA, AHB..)
  - Use cores from third party

![Diagram of the target architecture: Processor core1, IDCT, Memory, Processor core2, Prediction, core connected by an AMBA Bus.]
Codesign Methodology

- C Specification
- Thread Model
- Partitioning
  - S/W Synthesis
  - H/W Synthesis
- Co-simulation
- Prototyping
Partitioning

- **Hardware Estimation**
  - Input: Resources available
    - Functional units (type and number)
    - Memory (port and size)
  - Output: Number of cycles

- **Software Estimation**
  - Input: Instruction set architecture
  - Output: Cycles, program memory, data memory
Partitioning (IITD approach)

- SUIF based infrastructure
- Hardware estimation
  - Input
    - C function
    - HMDES: Machine description (FU, memory and latency)
  - Methodology
    - Data flow graph
    - List scheduling estimates

- Software Estimation
  - Architecture supported: ALPHA, PISA
Partitioning

- Application Profiling
  - GCC Compiler
    - Gprof: function level profiling
    - Gcov: line by line profiling
  - Intel’s compiler
    - VTune performance analyzer
Prototyping (Leon Based)

- **Leon**: A Sparc based microprocessor core
  - RTL level VHDL
  - Instruction slot available

- **Co-design Possibilities**
  - Hardware can access registers and memory
    - No communication overhead
    - Decoder design of processor effected
  - Hardware and processor has shared memory
    - Less communication overhead
  - Hardware has local memory and communication is by BUS
    - Communication overhead,
    - Least interference with Leon core
Prototyping (Commercial FPGA)

- **Xilinx support for Hardware-Software codesign**
  - Available core
    - Microblaze (Softcore)
    - PowerPC Hardcore
  - Software support
    - EDK (Embedded Design Kit)
    - Simulation and synthesis support for co-design
  - Instruction extension in microblaze is possible
  - Various cores are available with EDK

- **Altera support for hardware-software codesign**
  - NIOS based system
  - GUI support, Instruction extension to NIOS
Prototyping (Possible Student projects)

- **Image Processing**
  - Image kernels (DCT, equalization, histogram)

- **Scientific application**
  - Crypro – factoring algorithm
  - Linear equation solver