Introduction to Embedded Processors

CSE 291E / EE260C

Spring 2002

Overview

- Embedded Processors
  - Why?
  - Design Criteria
  - Architectural Options
- Example Architecture: ARM
  - Instruction Set Architecture
  - ARM7
  - ARM9
  - ARM10
- The Future

What is an Embedded Processor?

- An Embedded Processors is simply a uProcessors that has been “Embedded” into a device
- It is software programmable but interacts with different pieces of hardware – how?
- Performs both control and computation – more performance than a uController but not as much performance as a general purpose processor… yet
- Where are they used: Cars, Phones, Media Devices, Wireless, Printers – everyone uses them without thinking about it – start to think about it

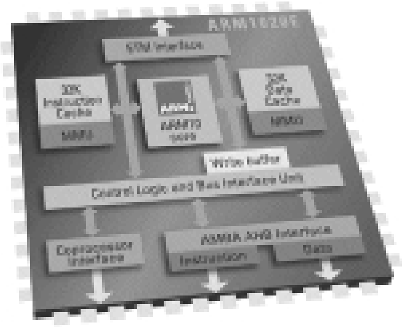
Some Places ARMs can be found

- Daewoo inet.top.box
- Bush Internet TV / box
- Datcom 2000 digital satellite receiver
- Pace digital satellite receiver (supplied as part of the Sky package)
- Numerous other digital cable / satellite receivers
- Hauppauge WinTV DVB-S PC TV card
- Oracle NC
- LG Java computer
- Millipede Apex Imager video board
- Paradise AiTV set top box
- Sony MZ-R90 minidisc
- Win-Jam
- JVC's digital camera 'Pixstar'
- Lexmark Z12/22/32/42/52 color Jetprinter
- Samsung office laser printer
- Samsung SmartJet MFP (printer/scanner/copier/fax)
- Xerox colour inkjet printer
- Digital logic analyzers from Controlware
- IHU-2 Experimental Space Flight Computer
- Siemens video phone
- Wizcom's Quicktionary
- Various GSM handsets, from the likes of Alcatel, AEG, Ericsson, Kenwood, NEC, Nokia...
- Cable/ADSL modems, by manufacturers such as Caymen Systems, D-Link, and Zoom.
- 3Com 3CD990-TX-97 10/100 PCI NIC with 3XP processor
- Routers, bus adaptors, servers, crypto, gateways...
- POS systems
- Smart cards
- Adaptec PCI to Ultra2 SCSI 64 bit RAID controller
- ATA drive electronics controller systems (bare)
- Iomega HipZip digital audio player
- C pen, with OCR and IrDA
- HP/Ericsson/Compaq pocket PCs
- Psion series 5 hand-held PC (5mx used 36MHz ARM710T)
- Various PDAs

What is it Really?

- Typically an Embedded Processor is a single-issue in-order RISC processor with a little cache
- It can then sold as a piece of silicon, custom layout, netlist, or architectural description
- They are designed to be small, low power, and most importantly correct.
- Often due to the real-time constraints of an application area they are designed to have a small deterministic worst case time per instruction – this is changing

ARM Chips

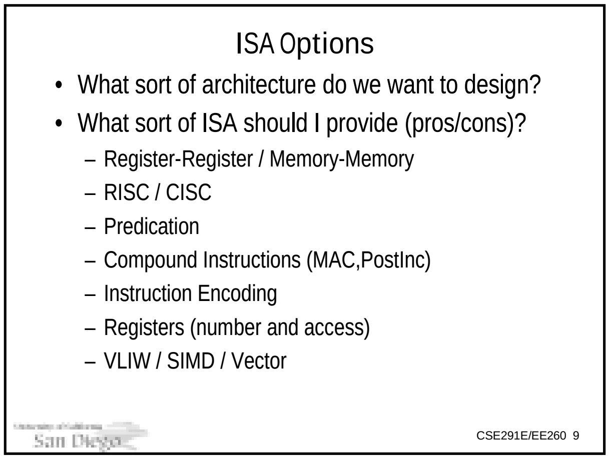


Why use an Embedded Processor?

- If I am John Q. RandomEngineer why would I want to build a system with an embedded processor built in?
- The main reason is simple: Cost
  - Embedded processors are small – so they don't take up much die area and thus they are cheap to fab
  - Embedded processors are verified – so I won't spend a bunch of engineering man hours traking down hardware bugs so I can tape out my chip
  - Embedded processors run software – the key part of that is the SOFT – deal with changing specs

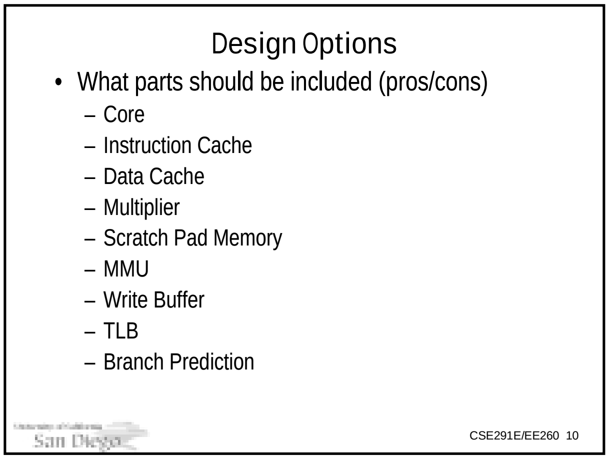
Design Criteria

- How do I design an "good" embedded processor?
- The three most important design criteria are performance, power, and cost.
  - Performance is a function of the parallelism, instruction encoding efficiency, and cycle time (or the good old NumInstr, CPI, Freq)
  - Power is approximately a function of the voltage, area, and switching frequency
    - Also a function execution time for leakage
  - Cost is a function of both area (how many fit on a die) and the complexity of use (in terms of engineering cost)



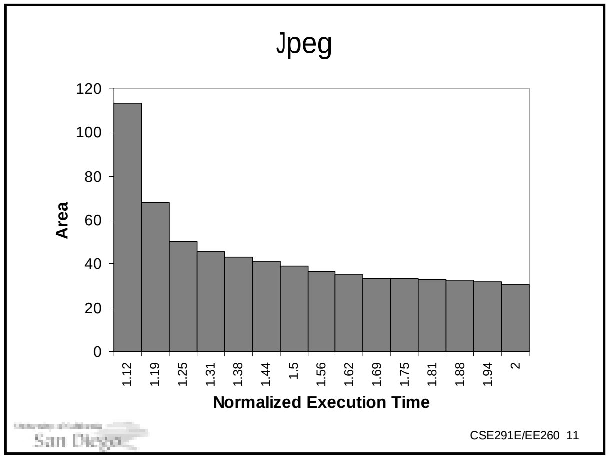
ISA Options

- What sort of architecture do we want to design?
- What sort of ISA should I provide (pros/cons)?
  - Register-Register / Memory-Memory
  - RISC / CISC
  - Predication
  - Compound Instructions (MAC,PostInc)
  - Instruction Encoding
  - Registers (number and access)
  - VLIW / SIMD / Vector

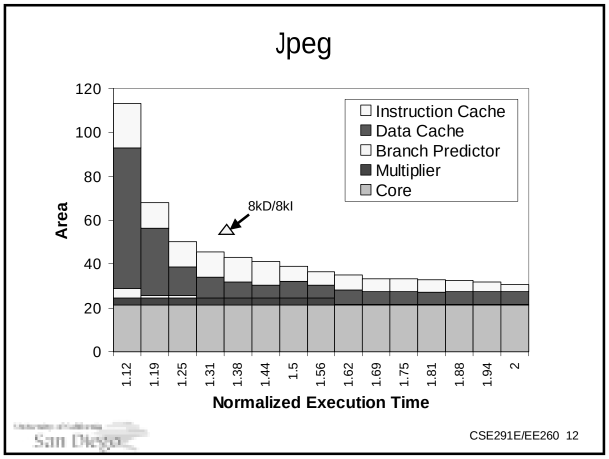


Design Options

- What parts should be included (pros/cons)
  - Core
  - Instruction Cache
  - Data Cache
  - Multiplier
  - Scratch Pad Memory
  - MMU
  - Write Buffer
  - TLB
  - Branch Prediction



Jpeg

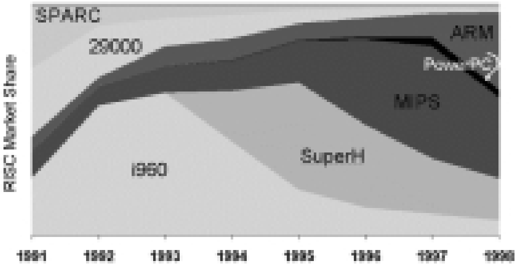


Jpeg

Example Architecture: ARM

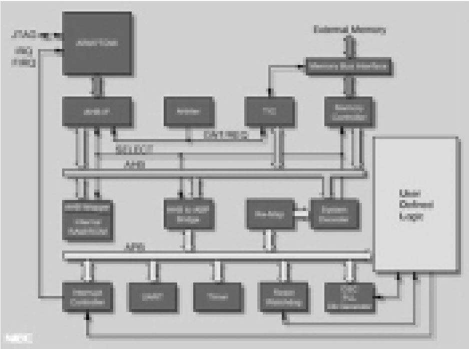
- – ARM licenses their core to companies as IP that you can drop into your SoC design
- – Other companies such as Intel license the ARM technology and build their own custom silicon
- – What are the design choices that ARM made?
  - ISA Design
  - Actual Implementation Details
- – 3 ARM Processors Families in production: ARM7, ARM9, and the recently released ARM10

RISC Processor Market Share



Source: vendors

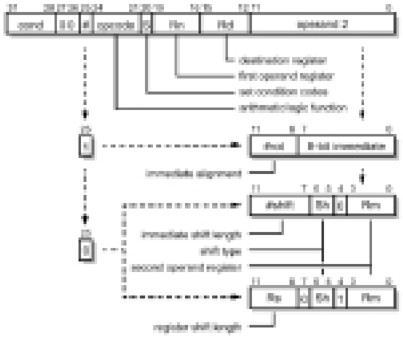
Example SOC



ARM ISA

- ARM is a Load-Store RISC Architecture
  - – First production RISC architecture ever
- 32-bit architecture
- All instructions are predicated
- 16 Registers
  - – r0-r14 are general purpose
  - – r15 is the program counter
- 32-bit instructions

ARM Instructions



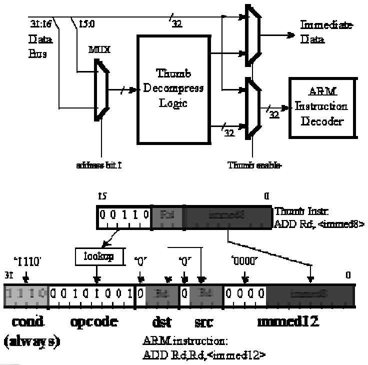
ARM Instructions

- Loads
  - Access can be byte, half-word, or word aligned
  - Lots of different indexing modes
  - Register indirect, Two register indirect, Register indirect with constant, Base+offset, Pre and Post increment
- Control
  - Control set up with comparison instruction (CMP)
  - Can be followed with a branch to a section of code
  - Can predicate following instructions
    - Using codes for equal, less than, overflow, carry set

Thumb Extensions

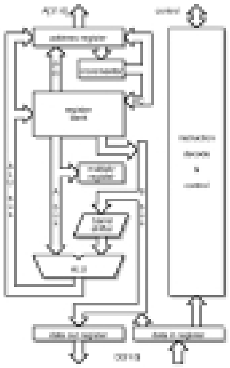
- First implemented in 1995 in the ARM7 core
- Thumb is a 16–bit subset of the ARM ISA
- It runs on a 32-bit chip so gets all of its benefits
- 32-bit address space, registers, shifter, ALU, memory transfer
- Thumb code is 65% of the size of ARM code,
- Lets software be designed for performance or code size on the granularity of a basic block
  - flexibility.

Thumb Decoder



ARM7 Data Path

- Two blocks shown, Data and Decode paths
- Two read port, one write port, additional ports for r15 (PC)
- Single cycle execute and write back



ARM7

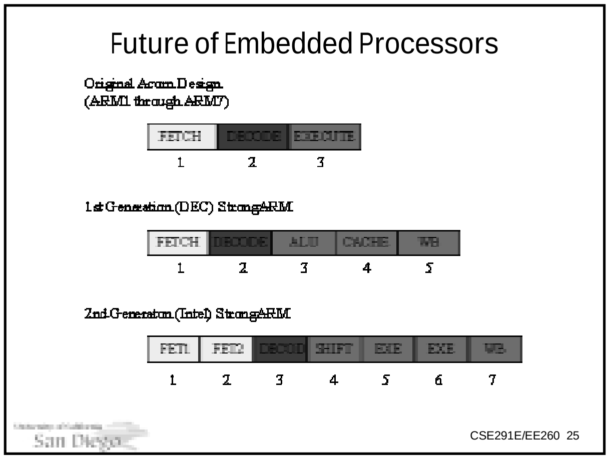
- Von-Neumann Architecture (8k cache)
- Simple 3 Stage Pipeline
- No penalty for unaligned access
  - Better for embedded applications
- In 0.13µm:
  - Die size of 0.26 mm^2
  - Greater than 133 MHz
  - IPC of 0.9
  - 0.06 mW/MHz

ARM9

- Harvard Architecture (8k Icache, 8k Dcache)
- 5 Stage Pipeline
- Improved MMU support
- 8 entry write buffer
- In 0.13µm:
  - Die size of 3.2 mm^2
  - Greater than 250 MHz
  - IPC of 1.1
  - 0.36/0.19 mW/MHz (with/without cache)

ARM10

- New 64-bit load-store architecture
- Up to 32K instruction and data cache
- 7 Stage pipeline
- New DSP instruction set
- Optional Vector Co-processor
- In 0.13µm:
  - Die size of 6.9 mm^2
  - Greater than 325 MHz
  - IPC of 1.25
  - 0.6 mW/MHz



Future of Embedded Processors

Original Acorn Design
(ARM1 through ARM7)

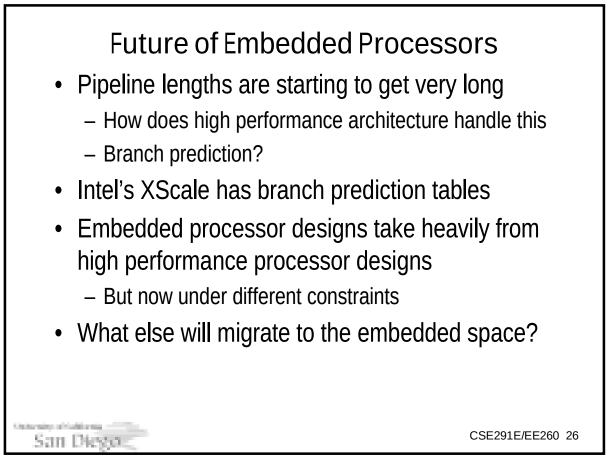
FETCH	DECODE	EXECUTE
1	2	3

1st Generation (DEC) StrongARM

FETCH	DECODE	ALU	CACHE	WB
1	2	3	4	5

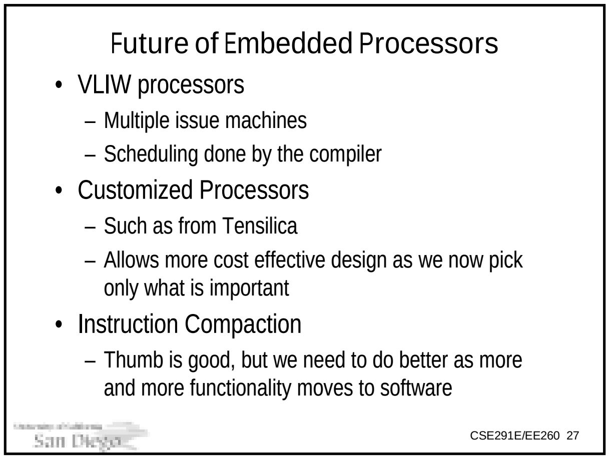
2nd Generaton (Intel) StrongARM

FET1	FET2	DECOD	SHIFT	EXE	EXE	WB
1	2	3	4	5	6	7



Future of Embedded Processors

- Pipeline lengths are starting to get very long
  - How does high performance architecture handle this
  - Branch prediction?
- Intel's XScale has branch prediction tables
- Embedded processor designs take heavily from high performance processor designs
  - But now under different constraints
- What else will migrate to the embedded space?



Future of Embedded Processors

- VLIW processors
  - Multiple issue machines
  - Scheduling done by the compiler
- Customized Processors
  - Such as from Tensilica
  - Allows more cost effective design as we now pick only what is important
- Instruction Compaction
  - Thumb is good, but we need to do better as more and more functionality moves to software