ARM integer cores

Outline:
- the ARM 3-stage pipeline
  - the ARM7TDMI core
  - the ARM 5-stage pipeline
  - the ARM9TDMI core
  - the ARM10TDMI core

The 3-stage ARM pipeline

- fetch
  - the instruction is fetched from memory
- decode
  - the instruction is decoded and the datapath control signals prepared for the next cycle
- execute
  - the operands are read from the register bank, shifted, combined in the ALU and the result written back

The 3-stage ARM pipeline

- Single cycle instructions
  - complete at a rate of one per clock cycle

The 3-stage ARM pipeline

- More complex instructions:

The 3-stage ARM pipeline

- PC behaviour
  - r15 increments twice before an instruction executes
    - due to pipeline operation
  - therefore r15 = address of instruction + 8 (+12 if used after first cycle, though this is architecturally ‘undefined’)
    - in Thumb code the offset is +4
  - normally the assembler makes the necessary adjustments, e.g. in branches

3-stage ARM organization

- ARM components:
  - register bank
    - 2 read ports, 1 write port
    - plus additional read and write ports for r15
  - barrel shifter
  - ALU
  - address register and incrementer
  - memory data registers
  - instruction decoder and control
The ARM7TDMI

- The ARM7TDMI is...
  - an ARM 3-stage pipeline core, with
  - T - support for the Thumb instruction set
  - D - support for debug
    - the processor can stop on a debug event
  - M - support for long multiplies
  - I - the EmbeddedICE macrocell
    - provides breakpoint and watchpoint hardware
    - described later

ARM7TDMI debug support
- the EmbeddedICE module
  - supports breakpoints and watchpoints
  - controlled via the JTAG test access port
- EmbeddedICE & JTAG are covered later

ARM7TDMI characteristics:

- Core area: 2.1 mm
- Power: 87 mW
- Clock: 0 to 66 MHz
- MIPS/W: 690
- Technology: 0.35 µm
- Transistors: 74,209
- Vdd: 3.3 V
- Multipliers: 0
- Cores: 2
- Processors: 1
- Power: 87 mW
- Clock: 0 to 66 MHz
- MIPS: 690
Outline:
- the ARM 3-stage pipeline
- the ARM7TDMI core
  ➔ the ARM 5-stage pipeline
- the ARM9TDMI core
- the ARM10TDMI core

Getting higher performance
- Increase the clock rate
  - the clock rate is limited by the slowest pipeline stage
    - decrease the logic complexity per stage
    - increase the pipeline depth (number of stages)
- improve the CPI (clocks per instruction)
  - fewer wasted cycles
    - better memory bandwidth

The 5-stage ARM pipeline
- Fetch
- Decode
  - instruction decode and register read
- Execute
  - shift and ALU
- Memory
  - data memory access
- Write-back

The 5-stage ARM pipeline
- Reducing the CPI
  - ARM7 uses the memory on nearly every clock cycle
    - for either instruction fetch or data transfer
  - therefore a reduced CPI requires more than one memory access per clock cycle
- Possible solutions are:
  - separate instruction and data memories
  - double-bandwidth memory (e.g., ARM8)

ARM9TDMI
- The ARM9TDMI is...
  - a ‘classic’ Harvard architecture 5-stage pipeline
    - separate instruction and data memory ports
  - with full support for Thumb and EmbeddedICE debug
  - aimed at significantly higher performance than the ARM7TDMI
    - enhanced pipeline operates at 100-200 MHz
ARM9TDMI pipeline

- Thumb instructions are decoded directly

ARM9TDMI

- EmbeddedICE
  - as ARM7TDMI, plus:
    - hardware single-stepping
    - breakpoints on exceptions
- On-chip coprocessor support
  - for floating-point, DSP, and so on

Process: 0.25 µm  Transistors: 111,000  MIPS: 220
Metal layers: 3  Core area: 2.1 mm²  Power: 150 mW
Vdd: 2.5 V  Clock: 0-200 MHz  MIPS/W: 1,500

ARM10TDMI

- The ARM10TDMI is...
  - aimed at significantly higher performance than the ARM9TDMI
  - achieved through use of:
    - higher clock rate
    - 64-bit I- and D-memory buses
    - branch prediction
    - hit-under-miss D-memory interface

ARM10TDMI pipeline

- Additional time allowed for
  - I- and D-memory accesses
  - instruction decode
- 6-stage pipeline