Introduction to the ARM Architecture

CM0506 – Small Embedded Systems

Dr Alun Moon

Department of Computer and Information Science

Lecture 3

RISC/CISC Processors

Complex Instruction Set Computers (**CISC**)

- Dense code, simple compiler
- Powerful instruction set, variable format, multi word
- Microcoded
- Multi-cycle execution

Reduced Instruction Set Computers (**RISC**)

- Small instruction set – hardwired
- High clock rate, low development costs
- Simple instructions, fixed format, complex optimising compiler
- Many registers may be banked
- Smaller die size, lower power

Is the ARM a RISC

- ARM has 16 GP registers (but so does the 68000)
- Some ARM instructions are complex:

```
    LDMEQFD r13!,{r0,r2-r4,pc}^
```

- ARM has simple hardware with modular design
- Smaller Si – quicker to market and easier to incorporate as IP
- ARM power consumption is low – battery operated kit, less cooling
- Most instructions executed in single cycle
- Access to memory is only through Load/Store instructions

ARM Design Philosophy

Variable cycle execution most instructions require a single cycle – some are longer

Inline barrel shifter expands many instructions – improved density and performance

Thumb 16-bit instructions improved code density for a subset of instructions

Conditional execution all instructions may be conditional – improved performance by reducing branching and increasing code density

Enhanced Instructions DSP instructions for 16x16-bit multiply

ARM Inside

ARM is one of the most popular 32-bit cores 4 billion ARM cores were manufactured in 2008

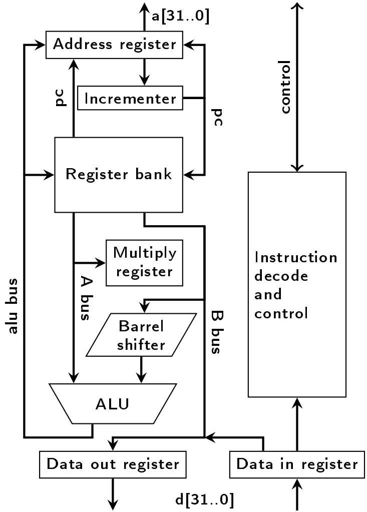
- Gameboy consoles
- Set-top boxes
- Satellite receivers
- WinTV
- Sony minidisc
- Digital cameras
- Laser printers

- GSM handsets
- Cable/ADSL modems
- Routers
- POS systems
- Smart cards
- Pocket PCs
- Hand-held PCs
- Various PDAs

The ARM Processor

- First ARM processor developed in 3 micron technology in '83-85'
- This course is based on the ARM7 architecture of early 90's
- DEC (now Compaq) developed the high performance StrongARM processor
- Recent developments are: ARM8 and ARM9E (1999), and a ARM processor without a clock – the AMULET
- We are using an ARM processor manufactured by NXP (Philips)

Internal Organisation of ARM7



- Two main blocks: datapath and decoder
- Register bank (r0 to r15)
- Two read ports for A-bus/B-bus
- One write port for ALU-bus
- Additional read/write ports for program counter r15
- Barrel shifter – shift/rotate 2nd operand by any number of bits
- ALU performs arithmetic/logic functions
- Address registers/incrementer holds PC address (with increment) or operand address

ARM Instruction Format

- 3 address format of ARM

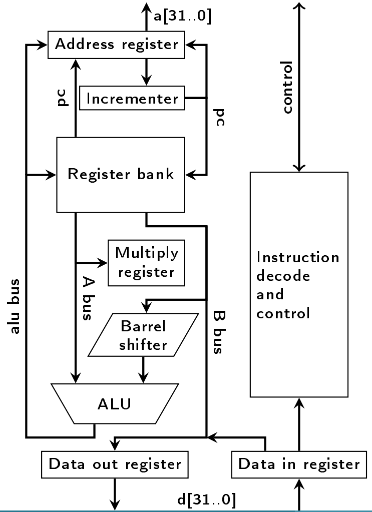
Function	Dest. addr.	Op 1 addr.	Op 2 addr.

- 2 address format of ARM Thumb instruction set

Function	Dest. addr.	Op 1 addr.

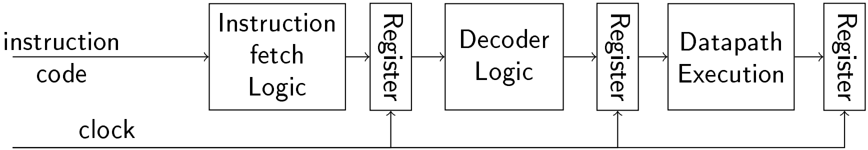
Instruction

Fetch/Execute cycle



- Data register holds read/write data to/from memory
  Instruction decoder decodes machine instructions to produce control signals to datapath
- In single-cycle data processing instructions, data values are read on the A-bus & B-bus, the result from the ALU is written back into register bank
- PC value in address register is incremented and copied back to r15 and the address register – allowing fetching new instructions ahead of time

Pipelining



- ARM7 uses 3-stage instruction pipeline

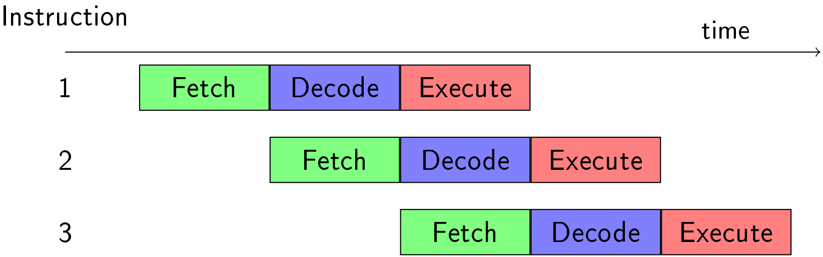
  Fetch fetch instruction code from memory into instruction pipeline

  Decode instruction decoded to obtain control signals for the datapath ready for the next stage

  Execute instruction “owns” the datapath – register read; shifting; ALU result generated and write-back

- Result of each stage stored in registers
- The consequence is that the clock period is much shorter than without pipelining

Pipelining



Instruction

time

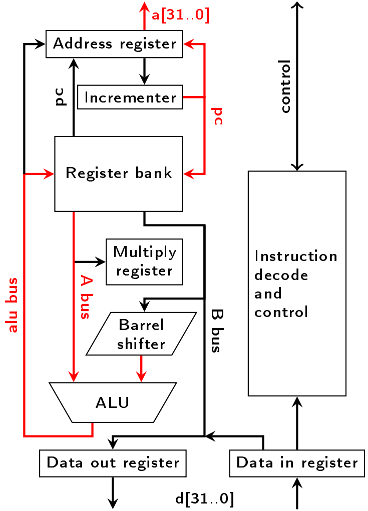
1 Fetch Decode Execute

2 Fetch Decode Execute

3 Fetch Decode Execute

- At any time, 3 different instructions may occupy each of the 3-stages of pipeline
- It may take three cycles to complete a single-cycle instruction – a three cycle latency
- Once a pipeline fills, the processor completes a single cycle instruction every clock cycle. Therefore the throughput is one instruction per cycle.

Datapath activity during data processing instruction



```
SUB r0, r1, #128 LSB #3
```

$$r_0 = r_1 - 128 \times 8 = r_1 - 1024$$

- Subtract instruction – one operand is a constant
- Constant 128 encoded in instruction passes through barrel shifter to produce 128*8
- ALU operates on the operands and writes the result back to r0
- PC value in address register is incremented and copied back to r15 and the address register