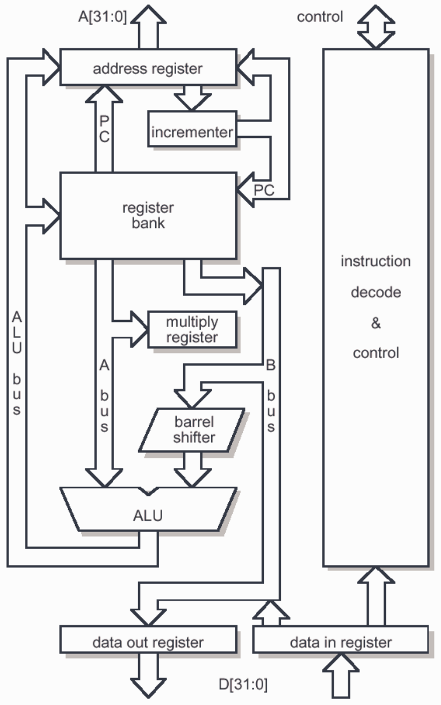
Lecture 7 ARM Processor Organization

- First ARM processor developed on 3 micron technology in ‘83-’85
- This course is mainly based on the ARM6/7 architecture developed between ‘90-’95.
- Digital Equipment Corporation (then Compaq, now HP) developed the StrongARM processor which has a very high performance.
- More recent developments are: ARM8 and ARM9E (1999), and a ARM processor without clock - the asynchronous AMULET from U. of Manchester (Steve Furber’s group)

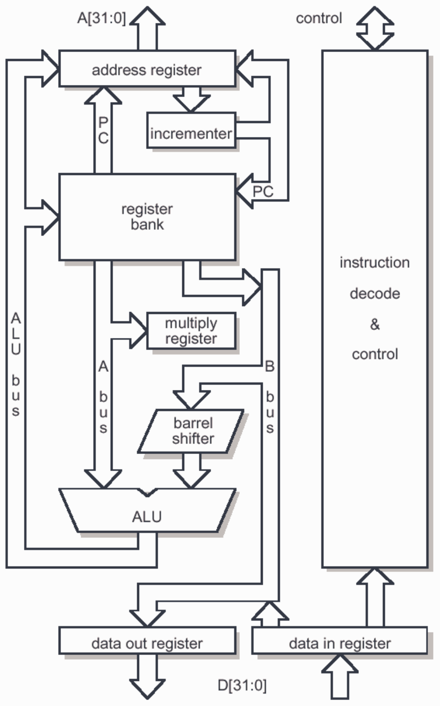
Internal Organization of ARM

- Two main blocks: datapath and decoder
- Register bank (r0 to r15)
  - Two read ports to A-bus/B-bus
  - One write port from 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
- Dedicated PC incrementer
- Address register – either from PC or from ALU



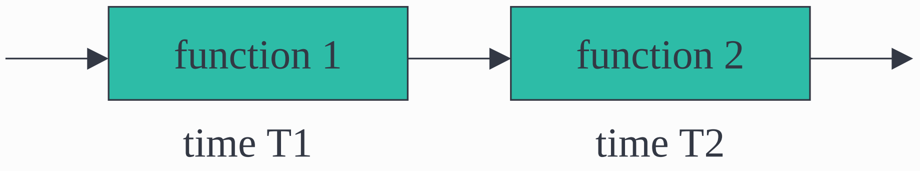
Internal Organization of ARM (con't)

- Data register holds read/write data from/to memory
- Instruction decoder decodes machine code instructions to produce control signals to datapath
- Data processing instructions take a single cycle: data values are read on the A-bus & B-bus, the results from ALU is written back into register bank

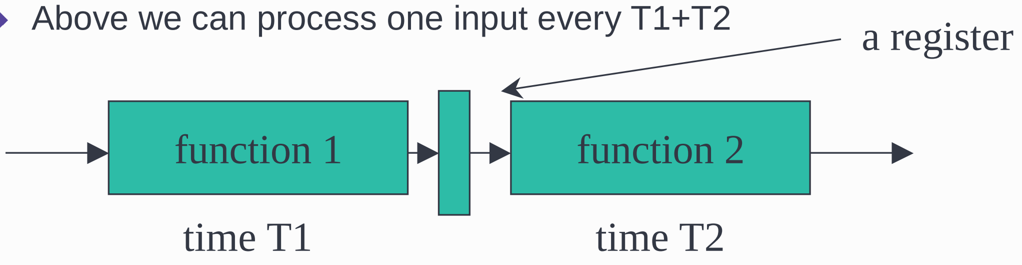


Pipelining

- The maximum processing rate is determined by the propagation delay of the computational logic – in abstract:



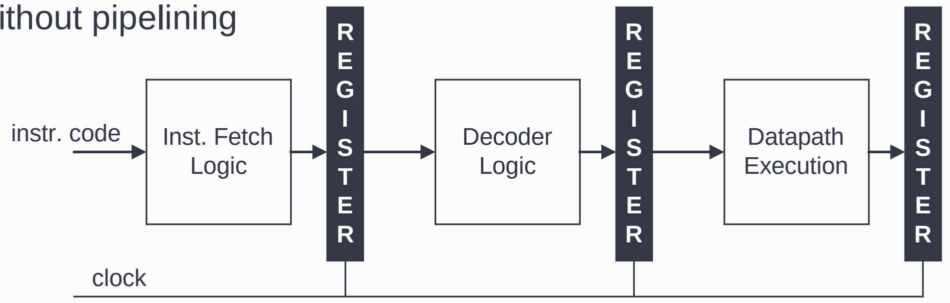
- Above we can process one input every T1+T2



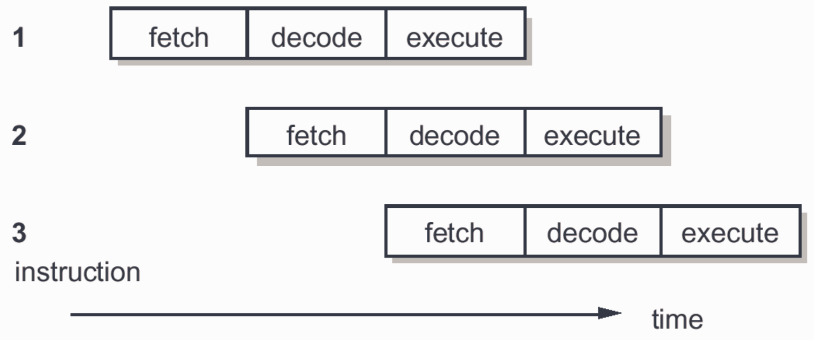
- Above we can process one input every max{T1,T2}, but each input still takes T1+T2 to be completely processed

Pipelining

- ARM uses a 3-stage instruction pipeline
  - **Fetch**: **fetch instruction code from memory into the 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 results generated and write-back**
- Results for each stage stored in registers
- The consequence is that the clock period is much shorter than without pipelining



Pipelining (con't)

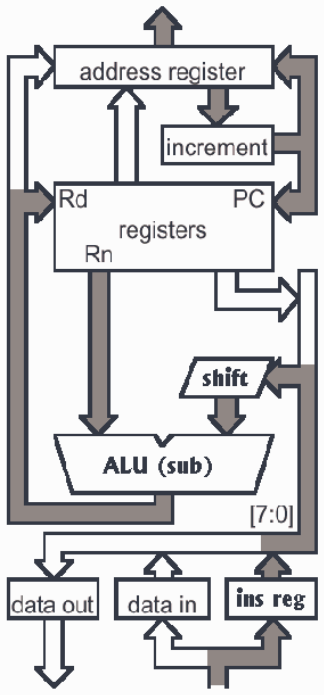


- At any time, 3 different instructions may occupy each of the the 3-stages of pipeline
- It may take three cycles to complete a single-cycle instruction. This is said to have a three cycle **latency**
- Once a pipeline fills, the processor completes a single-cycle instruction every clock cycle. Therefore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

Datapath activity during data processing instruction (register – immediate)

SUB r0, r1, #128, LSL #3 ; r0 := r1 - 128*8

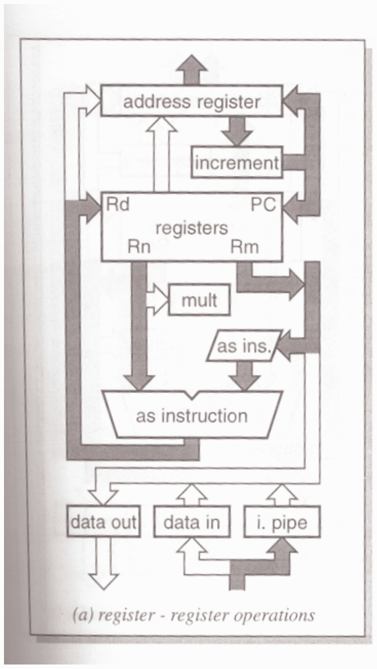
- Subtract instruction – one operand is a constant
- Constant 128 encoded in instruction passes through barrel shifter to produce 128*8
- Shifting can only be applied to the second operand!
- ALU operates on the operands and writes the result back to register r0
- PC value in address register is incremented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



Datapath activity during data processing instruction (register – register)

SUB r0, r1, r2, LSL #3 ; r0 := r1 – r2*8

- Subtract instruction – both operands are register-based
- Value of r2 and amount to shift by (encoded in instruction) passes through shifter to produce r2*8
- ALU operates on the operands and writes the result back to register r0
- PC value in address register is incremented and coped back to r15 and the address register



Summary

- ARM is fast for data processing instructions
  - throughput of 1 cycle per instruction
  - latency of 3 cycles per instruction
- Dedicated barrel shifter means a single data processing instruction can operate on a register and either:
  - a shifted immediate operand
  - a shifted register operand
- Two register operands are possible because register file has two read ports
- Dedicated incrementer means that ALU is not tied up with PC increment operations