COL788: Advanced Topics in Embedded Computing

Lecture 7 – Processor Architecture (Cont.)

Vireshwar Kumar
CSE@IITD

August 24, 2022

Semester I
2022-2023

Last Lecture

- ARM Registers
    - General purpose registers (r0-r14)
    - Dedicated program counter (r15)
    - Dedicated current program status register (cpsr)
    - Dedicated saved program status registers (spsr)
- Program Status Registers (cpsr and spsr)
    - Flags
    - Interrupt
    - Mode
- Calling operations with conditions

Arithmetic Operations

- Operations are:

ADD	operand1 + operand2
ADC	operand1 + operand2 + carry
SUB	operand1 - operand2
SBC	operand1 - operand2 + carry -1
RSB	operand2 - operand1
RSC	operand2 - operand1 + carry - 1

- Syntax:
  - <Operation>{<cond>}{S} Rd, Rn, Operand2
- Examples
  - ADD r0, r1, r2
  - SUBGT r3, r3, #1
  - RSBLES r4, r5, #5

Comparisons

- The only effect of the comparisons is to update the condition flags
    - Hence, there is no need to set S bit
- Operations are:
    - CMP   operand1 - operand2, but result not written
    - CMN   operand1 + operand2, but result not written
    - TST   operand1 AND operand2, but result not written
    - TEQ   operand1 EOR operand2, but result not written
- Syntax:
    - <Operation>{<cond>} Rn, Operand2
- Examples:
    - CMP   r0, r1
    - TSTEQ   r2, #5

Logical Operations

- Operations are:
  - AND		operand1 AND operand2
  - EOR		operand1 EOR operand2
  - ORR		operand1 OR operand2
  - BIC		operand1 AND NOT operand2 [ie bit clear]
- Syntax:
  - <Operation>{<cond>}{S} Rd, Rn, Operand2
- Examples:
  - AND		r0, r1, r2
  - BICEQ	r2, r3, #7
  - EORS	r1,r3,r0

Data Movement

- Operations are:
  - MOV   operand2
  - MVN   NOT operand2

  Note that these make no use of operand1.
- Syntax:
  - <Operation>{<cond>}{S} Rd, Operand2
- Examples:
  - MOV   r0, r1
  - MOVS   r2, #10
  - MVNEQ   r1,#0

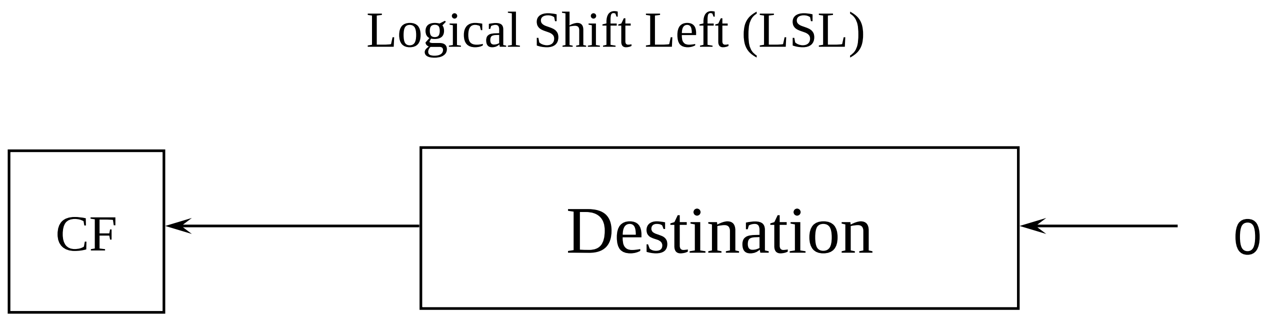
The Barrel Shifter

- The ARM doesn’t have actual shift instructions.

- Instead, it has a barrel shifter which provides a mechanism to carry out shifts as part of other instructions.

Barrel Shifter - Left Shift

- Shifts left by the specified amount (multiplies by powers of two) e.g.
  LSL #5 = multiply by 32

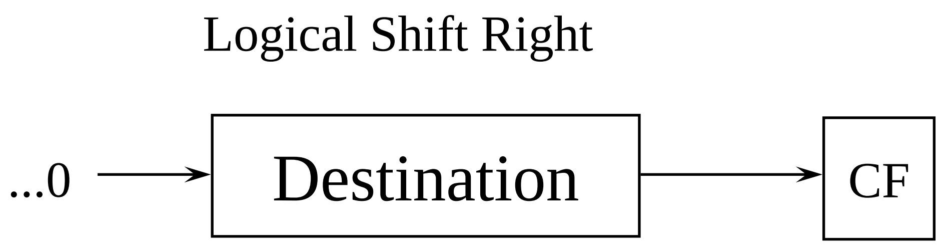


Barrel Shifter - Right Shifts

Logical Shift Right

- Shifts right by the specified amount (divides by powers of two) e.g.

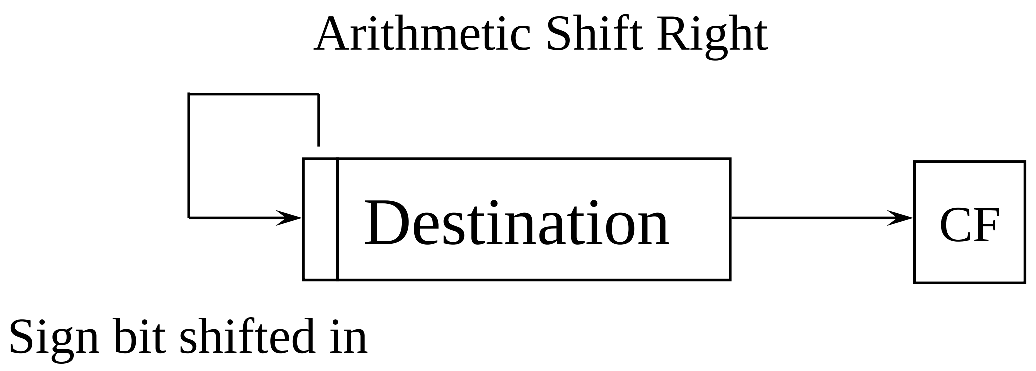
  LSR #5 = divide by 32



Arithmetic Shift Right

- Shifts right (divides by powers of two) and preserves the sign bit, for 2's complement operations. e.g.

  ASR #5 = divide by 32



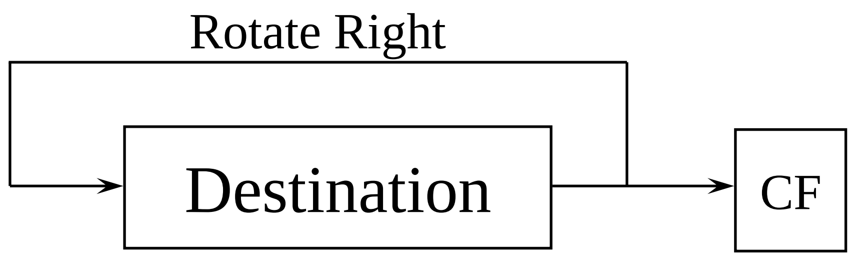
Barrel Shifter - Rotations

Rotate Right (ROR)

- Similar to an ASR but the bits wrap around as they leave the LSB and appear as the MSB.

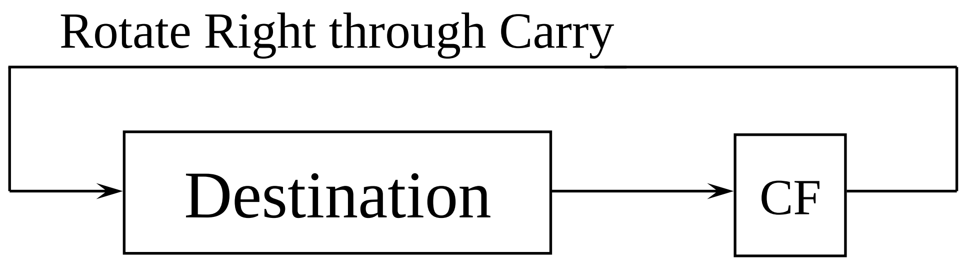
  e.g., ROR #5

- Note the last bit rotated is also used as the Carry Out.

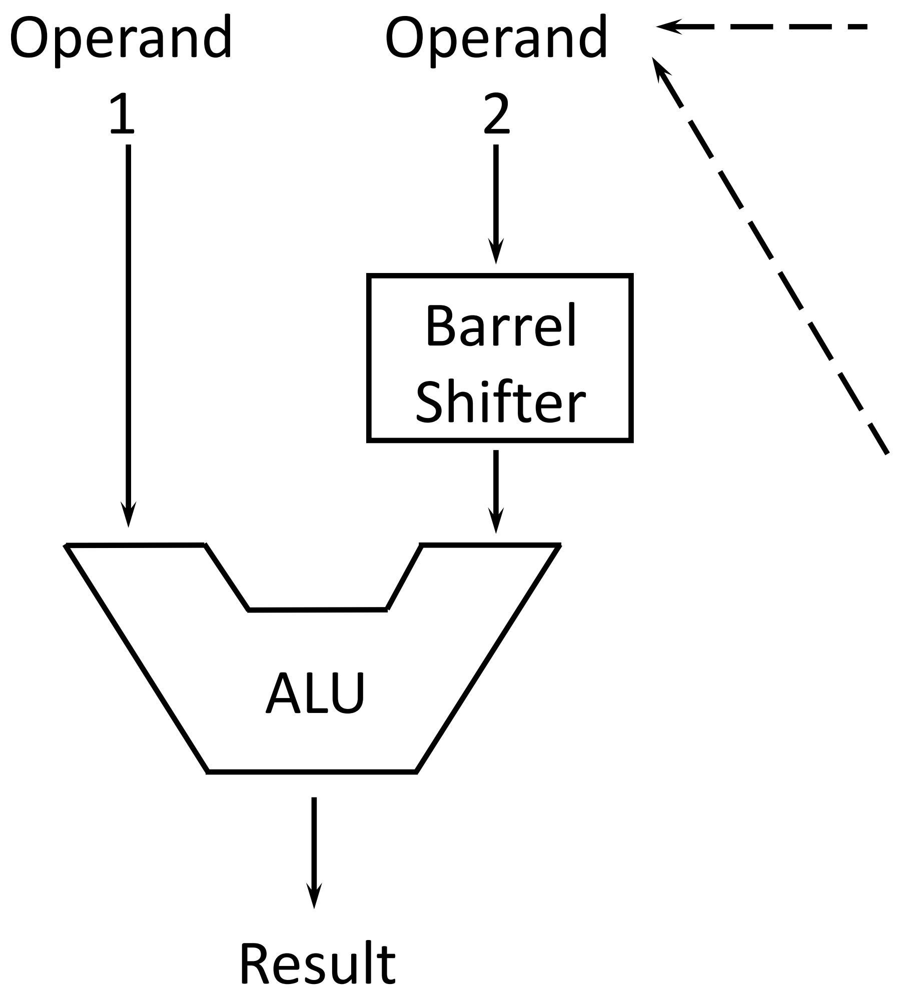


Rotate Right Extended (RRX)

- This operation uses the CPSR C flag as a 33rd bit.
- Rotates right by 1 bit. Encoded as ROR #0.



Using the Barrel Shifter: Second Operand



- Register, optionally with shift operation applied.
- Shift value can be either be:
  - 5 bit unsigned integer
  - Specified in bottom byte of another register.

* Immediate value

- 8 bit number
- Can be rotated right through an even number of positions.

Second Operand: Shifted Register

- The amount by which the register is to be shifted is contained in either:
    - the immediate 5-bit field in the instruction
        - No overhead as the Shift is done for free - executed in a single cycle.
    - the bottom byte of a register (not PC)
        - Then takes extra cycle to execute
        - ARM doesn’t have enough read ports to read 3 registers at once.
        - Then same as on other processors where shift is a separate instruction.
- If no shift is specified, then a default shift is applied: LSL #0
    - i.e., barrel shifter has no effect on value in register.

What’s Next?

- Next Lecture (August 25, Thursday, 12 pm – 1 pm)
  - Lecture 8