ARM Shifts and rotates
Addressing Modes
Fast Multiplication by constant

CSE 30: Computer Organization and Systems Programming

Diba Mirza
Dept. of Computer Science and Engineering
University of California, San Diego
Perform 1's complement on r0

A. EOR r0, r0, #0
B. MVN r0, r0
C. EOR r0, r0, 0xFFFFFFFFFF
D. BIC r0, r0, 0xFFFFFFFFFF
Shifts and Rotates

- **LSL** – logical shift by \( n \) bits – multiplication by \( 2^n \)

![Diagram showing LSL operation]

- C
- ...
Bit shifts and multiplication

LSL r0, r0,#4 is equivalent to which of the following expressions?

A. $r0 = 4 \times r0$
B. $r0 = 8 \times r0$
C. $r0 = 16 \times r0$
D. $r0 = r0/4$
Shifts and Rotates

- \text{LSR} – logical shift by n bits – unsigned division by \(2^n\)

\[0 \rightarrow \quad \ldots \quad \nearrow \quad \downarrow \quad C\]
Shifts and Rotates

- **ASR** – arithmetic shift by \( n \) bits – signed division by \( 2^n \)

![Diagram showing arithmetic shift](image)
What is the result in r0 after the following code is executed:

MOV r0, #-8
ASR r0, r0, #2

A. 0x4
B. 0xFC
C. 0xFFFFFFFFFC
D. 0xFFFFFFFFFE
Shifts and Rotates

- **ROR** – logical rotate by n bits – 32 bit rotate

![Diagram showing rotation of bits with...](image)

```plaintext
  ...  C
```

UCSD
Rotates in C

Which of the following C statements is equivalent to the given ARM statement

ROR r0, r0, #24  @Assume ‘int x’ is mapped to r0

A. x>>24
B. x>>24 | ((unsigned) x<<8)
C. (unsigned)x>>24 | x<<8
D. (unsigned)x>>8 | x<<24
# Syntax in ARM and C

<table>
<thead>
<tr>
<th>In ARM</th>
<th>In C</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSL</td>
<td></td>
</tr>
<tr>
<td>LSR</td>
<td></td>
</tr>
<tr>
<td>ASR</td>
<td></td>
</tr>
<tr>
<td>ROR</td>
<td></td>
</tr>
</tbody>
</table>
Addressing modes

- Register direct: all operands are registers
Addressing modes

- Immediate addressing: Last operand is an immediate
The data processing instruction format has 12 bits available for operand2

4 bit rotate value (0-15) is multiplied by two to give range 0-30 in steps of 2

Rule to remember is “8-bits rotated right by an even number of bit positions”
1. Register, optionally with shift operation
   ✤ Shift value can either be:
     ✤ 5 bit unsigned integer
     ✤ Specified in bottom byte of another register.
   ✤ Used for multiplication by constant

2. Immediate value
   ✤ 8 bit number, with a range of 0-255.
     ✤ Rotated right through even number of positions
   ✤ Allows increased range of 32-bit constants to be loaded directly into registers
Multiplication the fast way

Q: MUL r0, r2, #8; can be performed faster by which of the following ARM instructions

A. ADD r0, #0, r2, LSL #3
B. ADD r0, r2, r2, LSL #3
C. MOV r0, r2, LSL #3
D. None of the above
Multiplication the fast way

Q: MUL r0, r2, #7;

Write equivalent ARM expressions using ADD/SUB together with shifts and rotates
Add/Subtract instructions

1. ADD  r1, r2, r3;
2. ADC  r1, r2, r3;
3. SUB  r1, r2, r3;
4. RSB  r1, r2, r3;
Q: Write assembly code to put the last 12 bits of r0 into r1

A. AND r1, r0, 0xFFF
B. MVN r2, #0
   LSR r2, r2, #5
   AND r1, r0, r2
C. MVN r2, #0
   AND r1, r0, r2, LSR #5
D. MVN r2, #0
   AND r1, r0, r2, LSR #20
Conclusion

- Instructions so far:
  - Previously:
    ADD, SUB, MUL,
  - New instructions:
    RSB
    AND, ORR, EOR, BIC
    MOV, MVN
    LSL, LSR, ASR, ROR

- Shifting can only be done on the second source operand
- Constant multiplications possible using shifts and addition/subtractions