ARM Instructions II

- Move Instructions
  - Copies an immediate value or contents of a register into a destination register
  - MOV   Rd,Rm   ;   Rd ← [Rm]
  - MOV   Rd, #value   ; Rd ← value (8-bit immediate)
  - MVN   R0,#4   ; moves one's complement
    ; representation of 4 into R0
    - To move 2's-complement representation of c into a register, MVN can be used with c-1 as immediate operand
    - E.g. MVN R0,#4 places 2's-complement representation of -5 into R0   (0xFFFFFFFB is -4 as 1's-comp, -5 as 2's-comp)
    - Assembler treats MOV R0,#-5 as pseudo-instruction and substitutes MVN R0,#4

ARM Instructions II

- Implementing shift and rotate instructions
  - There are no explicit ARM shift or rotate instructions
  - Shifting or rotating source register in Move instructions produces the same effect.
    - MOV   Rd,Rm, LSL #4   shifts [Rm] left 4 bits and places result into Rd
    - MOV   R3, R3, ROR #16   ; swap left & right halfwords in R3

ARM Instructions II

- Logic instructions
  - AND, OR, XOR (bit-wise AND, OR, XOR)
  - Bit-Clear
    - BIC R0, R0, R1 ; where there is a 1 bit in R1, the
      ; corresponding bit in R0 is cleared
    - Works by ANDing 1’s-complement of [R1] with [R0]
- Bit Test instructions
  - TST   R2,#1 ; sets Z flag = 0, if LSB of R2 is 1 (non-zero)
    - ANDs [R2] with immediate value
  - TEQ R2,#6 ; sets Z = 1, if [R2] = 6
    - XOR’s the immediate value with the register

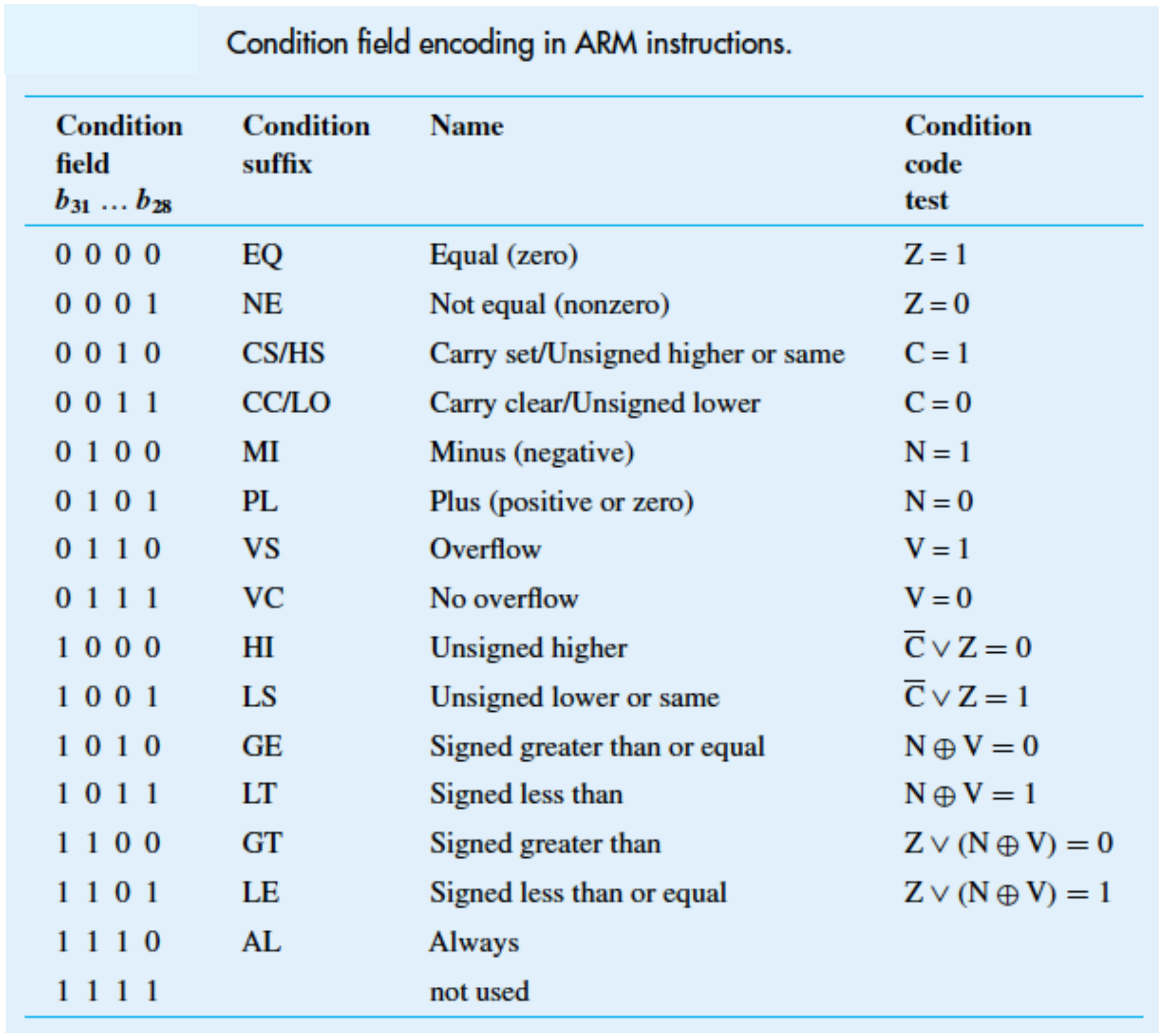
ARM Instructions II

- Compare instructions
  - CMP Rn,Rm   ; sets condition flags based on [Rn] – [Rm]
    - Result is discarded
    - Second operand may be an immediate value
  - CMN Rn,Rm ; sets condition flags based on [Rn] + [Rm]
    - Compare negative
    - Second operand may be an immediate value
  - Second operand may be shifted before use

ARM Instructions II

- Branch instructions
  - Conditional branch instructions contain a 24-bit two's complement branch offset
    - offset is shifted left 2 bits & sign-extended to 32 bits
    - Offset + updated PC = branch target address
    - Instruction condition field (bits 31 – 28) indicate test to be performed (branch is taken if condition is met)
  - Conditions are defined below:

ARM Instructions II



Condition field encoding in ARM instructions.

Condition field $b_{31} \ldots b_{28}$	Condition suffix	Name	Condition code test
0 0 0 0	EQ	Equal (zero)	$Z = 1$
0 0 0 1	NE	Not equal (nonzero)	$Z = 0$
0 0 1 0	CS/HS	Carry set/Unsigned higher or same	$C = 1$
0 0 1 1	CC/LO	Carry clear/Unsigned lower	$C = 0$
0 1 0 0	MI	Minus (negative)	$N = 1$
0 1 0 1	PL	Plus (positive or zero)	$N = 0$
0 1 1 0	VS	Overflow	$V = 1$
0 1 1 1	VC	No overflow	$V = 0$
1 0 0 0	HI	Unsigned higher	$\overline{C} \vee Z = 0$
1 0 0 1	LS	Unsigned lower or same	$\overline{C} \vee Z = 1$
1 0 1 0	GE	Signed greater than or equal	$N \oplus V = 0$
1 0 1 1	LT	Signed less than	$N \oplus V = 1$
1 1 0 0	GT	Signed greater than	$Z \vee (N \oplus V) = 0$
1 1 0 1	LE	Signed less than or equal	$Z \vee (N \oplus V) = 1$
1 1 1 0	AL	Always	
1 1 1 1		not used	

ARM Instructions II

- Subroutine Linkage instructions
  - Branch and Link instruction is used to call subroutines
    - BL SQRT   ; call routine with name SQRT
    - Return address (address of instruction after BL) is loaded into R14 (the link register)
    - Link register must be saved on stack before a nested call
    - R13 is used as the processor stack pointer
  - STMFD is used to save (push) registers onto stack
    - Suffix FD means predecrement R13 toward lower addresses
  - LDMFD is used to restore (pop) contents of registers
  - Restoring PC (R15) returns to calling routine
    - MOV   R15,R14   ; copy link register into PC to return