Reduced ARM ISA

Storage space

In this exercise, ARM uses

- 4 Gigabytes (2^{32} bytes) of memory
- A register file of 16 registers of 32 bits each. They are called r0, r1, up to r15.

Reduced instruction set description

Transfer instructions			
Instruction	Mnemonic	Operation	Example
Load word	ldr rd, [rn, #offset]	rd ← mem[rn + offset]	ldr r5, [r0, #32] r5 ← mem[32+r0]
Store word	str rd, [rn, #offset]	mem[rn + offset] ← rd	str r4, [r2, #16] mem[16+r2] ← r4
Move	mov rd, rn	rd ← rn	mov r3, r4 r3 ← r4
Move immediate	mov rd, #imm	rd ← imm	mov r3, #15 r3 ← 15
Arithmetic-logical instructions			
Instruction	Mnemonic	Operation	Example
Add Immediate	add rd, rn, #imm	rd ← rn + imm	addi r5, r2, #-15 r5 ← r2 + (-15)
Add	add rd, rn, rm	rd ← rn + rm	add r2, r3, r4 r2 ← r3 + r4
Subtract	sub rd, rn, rm	rd ← rn - rm	sub r4, r5, r6 r4 ← r5 - r6
And Immediate	and rd, rn, #imm	rd ← rn Λ imm	andi x5, x2, #2 r5 ← r2 Λ 2
And	and rd, rn, rm	rd ← rn Λ rm	and r2, r3, r4 r2 ← r3 Λ r4
Or Immediate	or rd, rn, #imm	rd ← rn v imm	ori r5, r2, #2 r5 ← r2 v 2
Or	or rd, rn, rm	rd ← rn v rm	or r2, r3, r4 r2 ← r3 v r4
Shift Left Logical Immediate	lsl rd, rn, #imm	rd ← rn ≪ imm	lsl r2, r3, #2 r2 ← r3 ≪ 2
Shift Right Logical immed	lsr rd, rn, #imm	rd ← rn ≫ imm	lsr r2, r3, #2 r2 ← r3 ≫ 2

Control and comparison instructions			
Instruction	Mnemonic	Operation	Example
Compare	cmp rd, rn	**flags** ← flags(rd - rn)	cmp r1, r0
Branch Equal	beq label	if **EQcond**(**flags**) then pc ← label	beq loop
Branch Less Than	blt label	if **LTcond**(**flags**) then pc ← label	blt loop
Jump	b label	pc ← label	b loop
Jump and Link	bl label	r14 ← pc+4, pc←label	bl loop
Jump Register	bx rd	pc ← rd	bx r14

The ARM processor stores a number of **flags** (flip-flops) that as conditions after executing some specific arithmetical-logical operations. Flag Z asserts if the result of an operation has been 0, flag N when the result is negative, flag C on carry out etc. Executing operations between values, those flags can be set. Control instructions **EQcond**(**flags**) test equality in a former instruction (via subtract). **LTCond**(**flags**) tests one has been smaller than the other.

Exercise

Given a list of common actions in assembly programming. Give for each action (a list of) instruction which would make it work. Describe the operands that are used.

Load a specific number into a register → Load 5 into register 5
Move the content of one register to another → Move the content of register 0 to register 6
Load a memory position content into a register → Load mem[64] into register 4
Store a register content into a specific memory position → r3 content in memory position 32