

Typical Questions for Experiment 2

Checkpoint 1

- What are the addresses of variables *a*, *b* and *c*?
- What is the meaning of the instruction “`ldr r1, [pc, #12]`” in program *add-v1.s*?
- What is the significance of `#12` in the instruction “`ldr r1, [pc, #12]`”?
- What is the meaning of the directive `.word`?
- What is the procedure for swapping variables *a* and *b*?

Checkpoint 2

- Why is the instruction “`ldr r1, a`” called a *pseudo-instruction*?
- In the instruction “`ldr r1, [pc, #offset]`”, what is the range `#offset`?
- What is the difference between the instruction “`ldr r1, a`”, and the instruction “`ldr r1, =a`”?
- What is the purpose of the literal pool?
- What is the meaning of the instruction “`add r1, r2, r2 lsl #2`”?

Checkpoint 3

- Identify the instructions in the program *exp.s* in Figure 9 that correspond to the C statement “`(k = 0; k < 31; k++)`”.
- What is the meaning of the compiler options `-O1` and `-O2`?
- In the program *gcd.c* in Figure 10, how many times will while loop be executed, if *a* = 23 and *b* = 17?
- Identify the instructions in the your program of *gcd.s* that correspond to the C statement “`while (a != b)`”.

Checkpoint 4

- How do you implement C’s “`if`”, “`else`” and “`while`” in the Assembly language?
- Why do you assign the C variable `multiplcand` to register `R0` the variable `multiplier` to `R1` and return result in `R0`?

Checkpoint 5

- The relation “ $f(y) = a + (y \times b)$ ” involves the variable *y*. Why?

Checkpoint 6

- Why is the positional multiplication more efficient?
- Why does the positional multiplication start from the most significant bit?
- In you assembly program of *posn-mul.s* where do you keep the constant values `m7 - m1`?

Checkpoint 7

- What is the number of the executed assembly language instructions if all of the `if`-statements are true?
- What is the number of the executed assembly language instructions if none of the `if`-statements are true?

Checkpoint 8 (No Credit)

- Why should the greater common divisor be always expressed as a positive number?
- What is the greater common divisor of -123/23?

Checkpoint 9 (No Credit)

- How much bigger your assembly code is compared with your program of *gcd.s*?

Checkpoint 10 (No Credit)

- How did you deal with the most significant bit?

Checkpoint 11 (No Credit)

- How much bigger your assembly code is compared with your program of *posn-mul.s*?