LUT University

Computational Engineering 2022-01-21

BM40A0202 Foundations of Computer Science

Toni Kuronen

Exercise 4 (week 6), Micro-programming

Tasks (1p/task)

1. The micro-programmable computer's micro-program memory (MPM) contains the program below. What happens step by step during the execution of the program and what is the purpose of the program?

   Binary microcode:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
0				1										1	1		1					1
1						1			1								1					1
2					1			1			1						1					1
3			1		1									1	1		1					1
4	1					1				1							1					1

2. Write a symbolic microprogram and corresponding binary microcode (l. 22-bit commands) when one wants to calculate a sum $A = \sum_{i=0}^{D} i$, where A refers to the register A and where the value limiting the number of repetitions is in central memory in the memory location indicated by register D.

   The starting point is the pseudo-code of the program written in the previous exercises.

3. How does the computer handle floating-point numbers? Floating-point numbers are commonly represented in the form $u = (e, f) = f \times b^{e-q}$, where b is the system's base number, q is the scaling factor, and $|f| < 1$.

   For two floating point numbers $u = (e_u, f_u)$ and $v = (e_v, f_v)$, $u > v$, the summation $u + v = w$, $(e_u, f_u) \oplus (e_v, f_v) = (e_w, f_w)$ is in principle of the form

$$e_w = e_u, f_w = f_u + f_v / b^{e_u - e_v}$$

   (a) What kind of an algorithm performs the addition?

   (b) How do you estimate the workload of the floating point addition compared to an integer addition?

   (c) Assume that $b = 10$ and the presentation accuracy of the floating point number p is $p = 8$. Test your algorithm (calculate the sums $w = u_1 + v_1$ and $w = u_2 + v_2$) with the base 10 numbers $u_1 = (50, +0.98765432)$, $u_2 = (45, -0.50000001)$ and $v_1 = (49, +0.33333333)$, $v_2 = (54, +0.10000000)$.

PO BOX 20, FI-53851 LAPPEENRANTA, FINLAND
E-mail: toni.kuronen@lut.fi

4. Modified school algorithm for the microprogrammable computer presented below as a pseudo-code algorithm. The dividend is stored in register A, the divisor in register MDR (main memory data register), and the integer part of the quotient is stored in register C. The algorithm adds trailing zeros to the divisor so that it becomes greater in value than the divident.

   (a) Perform a division calculation 109/11 with binary numbers using the modified school algorithm showing all the intermediate steps.

   (b) From what does the number of iterations in the algorithm depend on?

```
D := 0            (* number of digits in the quotient *)
WHILE MDR <= A DO (* calculate D *)
C := MDR
MDR := 2 * MDR
D := D + 1
ENDWHILE
MDR := C          (* the divisor complemented by trailing zeros *)
C := 0            (* initial value of the quotient *)
WHILE D > 0 DO    (* calculate the quotient *)
C := 2 * C        (* add a number to C *)
IF MDR <= A THEN
C := C + 1        (* the new number is 1 *)
A := A - MDR      (* the divisor is subtracted from the divident *)
ENDIF
A := 2 * A        (* the divident is moved *)
D := D - 1
ENDWHILE
```