

## Exercise Sheet 11 (Block C - 3)

(16 points)

Submission until Wednesday, 25th January 2017, 16:00 o'clock  
Discussion begins on Monday, 30th January 2017

Please see notes at the end of the document for the submission.

### 11.1 N-address machines (4 points)

Disassemble the instruction  $E := A + (B - C) * D$  in a command sequence for:

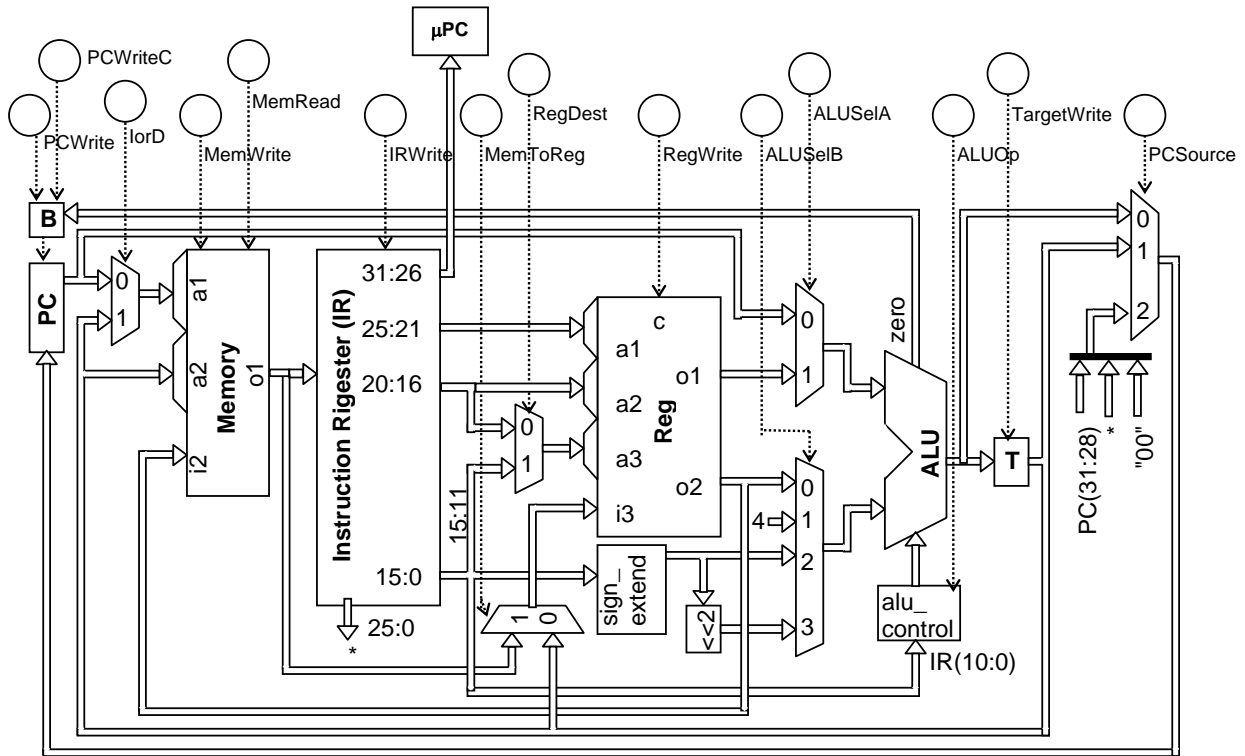
- 3-address machine
- 2-address machine
- 1-address machine

Write the command sequences with the minimum possible commands. The contents of the variables  $A, \dots, D$  cannot be overwritten. Use as few auxiliary variables as possible.

### 11.2 Control signals and data paths (4 points)

The following hardware structure was presented in the lecture<sup>1</sup>, which is designed to run the MIPS instruction set. The operations for a *store*-command should be executed in the instruction execution phase. The control unit is already in the *store* state.

Identify the used data paths for the execution of all operations by coloring them. Also, write all the control signals into the corresponding circles at the top of the digram according to the notations used in the lecture.



<sup>1</sup><http://ls12-www.cs.tu-dortmund.de/daes/media/documents/teaching/courses/ws1617/rs/rs2.3-microarch.pdf>

### 11.3 Datenabhängigkeiten (4 points)

- a. The commands B1, ..., B6 are executed sequentially. For each command, specify on which other commands it is **directly or true data dependent** (*read-after-write*).

B1: **add** \$2, \$2, \$3  
 B2: **sub** \$2, \$2, \$4  
 B3: **and** \$4, \$2, \$5  
 B4: **mul** \$5, \$5, \$7  
 B5: **mul** \$4, \$2, \$4  
 B6: **div** \$6, \$8, \$5

	B1	B2	B3	B4	B5	B6	not dependent
B1 is data dependent on							
B2 is data dependent on							
B3 is data dependent on							
B4 is data dependent on							
B5 is data dependent on							
B6 is data dependent on							

- b. The commands B1, ..., B6 are executed sequentially. For each command, specify on which other commands it is **anti-data-dependent** (*write-after-read*).

B1: **add** \$2, \$3, \$5  
 B2: **and** \$4, \$2, \$5  
 B3: **xor** \$5, \$4, \$2  
 B4: **add** \$2, \$3, \$4

	B1	B2	B3	B4	not dependent
B1 is anti-data-dependent on					
B2 is anti-data-dependent on					
B3 is anti-data-dependent on					
B4 is anti-data-dependent on					

- c. Which commands from sections (a) and (b) have output dependency (*write-after-write*).

## 11.4 System calls and assembler programming (4 points)

In dieser Aufgabe sollen Sie Ihre Implementierung der Fakultätsfunktion von Aufgabe 9.3 wiederverwenden<sup>2</sup> und um eine Benutzereingabe ergänzen. Der Benutzer soll die Zahl eingeben können, statt dass sie im Programm mit `ein: .word 5` fest vorgegeben ist. Schreiben Sie dazu ein Assemblerprogramm, welches den Benutzer mit einem kurzen Text auffordert, eine Zahl einzugeben, deren Fakultät berechnet werden soll. Nach Eingabe der Zahl soll das Programm dann deren Fakultät ausgeben. Ändern Sie für die Berechnung der Fakultät den Code von Aufgabe 9.3 zu einer aufrufbaren Funktion gemäß den MIPS-Konventionen (siehe Aufgabe 10.2<sup>3</sup>) ab.

Nutzen Sie für die Ein-/Ausgabe Ihres Programms Systemaufrufe. Die Aufrufe mit den Nummern 5 und 1 unterstützen in MARS das Ein-/Ausgeben von Ganzzahlen. In der MARS Hilfe finden Sie die detaillierte Dokumentation zu den einzelnen Systemaufrufen. Beenden Sie Ihr Programm wie üblich mit einem Systemaufruf mit der Servicenummer 10.

Benutzen und ergänzen Sie folgendes Programmsegment:

```
.data
aufforderung: .asciiz "Berechne Fakultät von:_" # Eingabeaufforderung (1)

.text
.globl main
main:
```

---

<sup>2</sup><http://ls12-www.cs.tu-dortmund.de/daes/media/documents/teaching/courses/ws1617/rs/blatt09.pdf>

<sup>3</sup><http://ls12-www.cs.tu-dortmund.de/daes/media/documents/teaching/courses/ws1617/rs/blatt10.pdf>

### Notes:

Submission until Wednesday, 25th January 2017, 16:00 pm in the mailbox number 40 at Otto-Hahn-Straße 12.

You can find the mailboxes in the first floor of the Otto-Hahn-Straße 12 near the transition to the ground floor of the Otto-Hahn-Straße 14. The mailboxes are labeled with "Rechnerstrukturen", the exercise group number and time/place of the exercise. The English exercise group is number 31 and the mailbox is number 40.

Please write your **name**, your **student registration number** and your **exercise group number** at the top right corner of your submission. You can make submissions in teams with up to two more students. To make a team submission put names, student registrations numbers and group numbers of all members of the team on the submission. Only one submission per team has to be made.

Tack you submission. Please do not fold your submission and do not put it into an envelope. Use the correct mailbox, you will need your exercise group number for that.

In total there are 12 exercises in 3 blocks (A, B, C). In each block you have to achieve at least 30 points of 64 possible ones to get access to the exam.