Master Parisien de Recherche en Informatique
2017-2018

Systèmes Synchrones
T. Bourke, M. Pouzet & J. Vuillemin

# TP1: Hack Processor

The goal of this exercice is to program in Lustre the processor Hack defined in the book "The Elements of Computing Systems" by Noam Nisan and Shimon Schocken (web site: `www.idc.ac.il/tecs`).

## 1 Installing Lustre

**Question 1**
*The first thing is to install Lustre.*

To compile a Lustre file with name `file.lus` and main node to simulate is `main`, you must compile Lustre in ec code:

```
--> lus2ec file.lus main
```

Then, file ec is compiled to C (option `-loop` generates a main loop which call node `main`):

```
--> ec2c main.ec -loop
```

Finally, compile the C code:

```
--> gcc -o main main.c main_loop.c
```

Is is also possible to simulate a Lustre program with the graphical interface `luciole`:

```
--> luciole file.lus main
```

Finally, if the file `file.lus` contains a node `check` with a single output of type `bool`, you can try to prove that the output is always true with tool `lesar`:

```
--> lesar file.lus check -v -diag
```

## 2 Implementation of the Hack processor

The specification of every element of the processor Hack is given in the book "The Elements of Computing Systems" available at:

`http://www1.idc.ac.il/tecs/plan.html`

**Question 2**
*The Lustre program to complete can be found at:* `http://www.di.ens.fr/~pouzet/cours/mpri/tp1/hack.tar.gz`*.*

**Question 3**
*Implement the node* `DMux` *(cf. chapter 1, page 21).*

**Question 4**
*Prove that the composition of a demultiplexer and a multiplexer is equivalent to the identity function Prouver que la composition d'un démultiplexeur et d'un multiplexeur if the two node have the same select condition.*

**Question 5**
*Define a new implementation of node* `or_n_way` *(cf. chapter 1, page 23) which uses a logarithmic number of* `or` *gates with respect to its number of inputs (we suppose that the number of input is a power of 2).*

**Question 6**
*Prove that this new implementation of node* `or_n_way` *is equivalent to the previous implementation in case* `n` *equals 8.*

**Question 7**
*Implement the $n$-bits adder* `add_n` *by using the node* `FullAdder` *(cf. chapter 2, page 34). Use it to implement a 16-bits adder.*

**Question 8**
*Implement the arithmetic and logic unit* `ALU` *(cf. chapter 2, page 36). To test your ALU, use nodes* `check_ALU*` *from file* `check.lus`*.*

**Question 9**
*Implement the node* `Bit` *(cf. chapter 3, page 48).*

**Question 10**
*Implement the node* `PC` *(cf. chapter 3, page 51). To test the behavior of your PC, use the node* `check_PC` *from file* `check.lus`*.*

**Question 11**
*Implement the CPU of processor Hack (cf. chapter 5, page 88). You can follow the architecture description given in chapter 5, page 94. To determine the value of control bits, see the description of instructions in chapter 4, pages 64 à 69. You can test the CPU with the node* `incr_machine` *which compute the sequence of integers modulo $2^{16}$.*