# Microprocessor design

Baptiste Lefebvre, Li-yao Xia, Antonin Delpeuch

18 Janvier 2013

## 1 Motivation

We wanted to design an original microprocessor. The requirements were :
- To be able to run a digital watch ;
- To use an uncommon architecture, contrasting with the MIPS standard as we already discovered it in the compilation course ;
- To be able to run arbitrary programs.

Reduced Instruction Set Chips (such as ARM) are widely used, but their instruction set is not really small. A natural question arises : to what extent can we reduce this set ?

## 2 Instruction set

We discovered the Subleq instruction set, which is made of only one instruction, called `subleq`. Here is the behaviour of `subleq a b c` :

```
b := b - a;
if b > 0 then
  go to the next instruction
else go to c
```

With this only instruction, one can compute the usual operations and the chip remains therefore able to run arbitrary programs (as long as it has access to enough memory).

As the chip always handle signed values, we decided that writing (respectively reading) from a negative address sets (respectively returns) the value of the given output (which is a value displayed on 7 segment modules).

An instruction invariably takes 5 clock ticks to be executed, because we use one single addressed RAM. Less clock ticks could be used through the use of parallel addressing, which could be implemented but wasn't because we deemed it too artificial with the basic blocks provided to us. It was also an opportunity to "schedule" the execution of an instruction over several time steps.

We also chose to implement the conversion from binary values to 7-segment codes in the circuit.

## 3 Logic circuit simulator

For this project we had to write a simulator. Advanced optimizations were not required of us, but we implemented an optimizing simulator, which looks for "common sub-expressions" and partially evaluates constant values.

## 4 Development tools

We implemented an assembler and an assembly language interpreter which defines the behaviour we expect from our circuit.

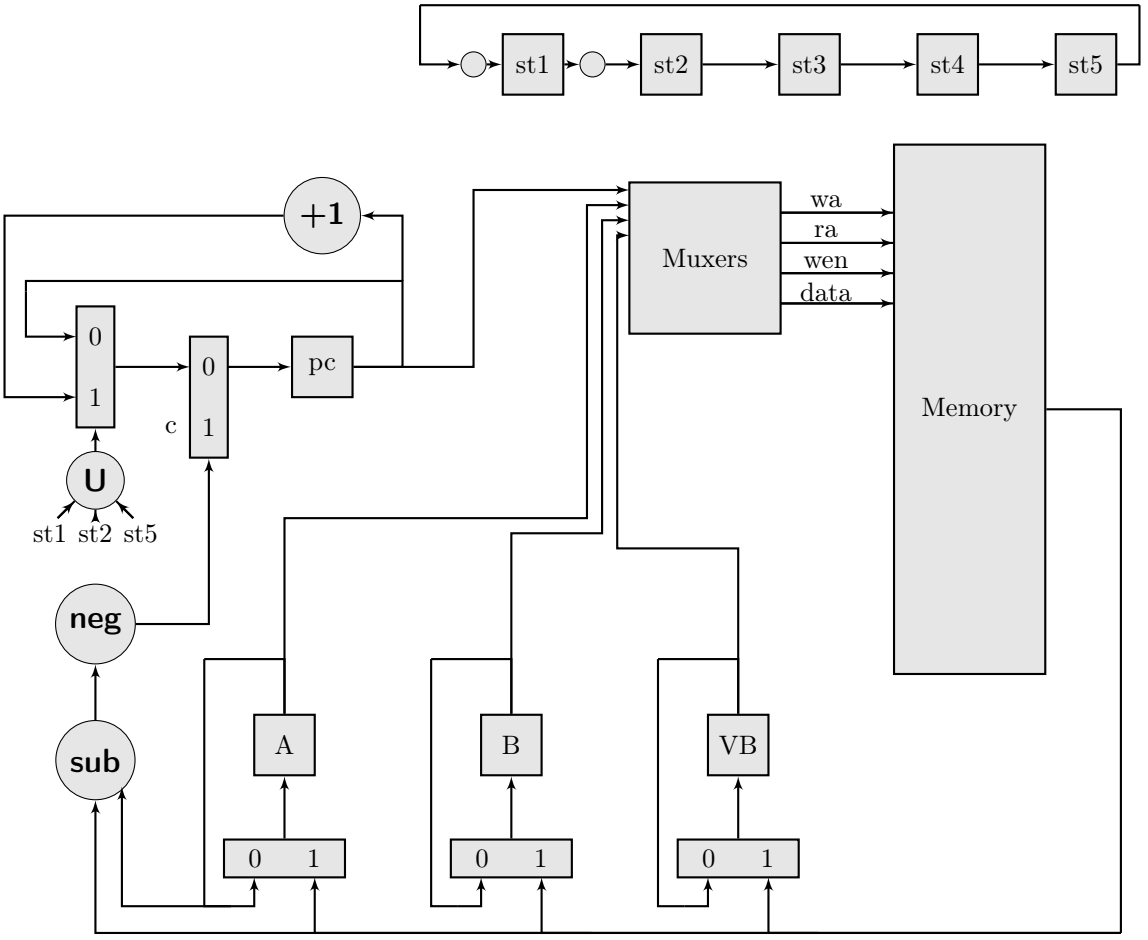


FIGURE 1 – Chip's architecture

## 5 Watch

We wrote the program of our watch in the subleq assembly language. The program implements the Gregorian calendar including leap years (every four years except every 100 years except every 400 years)