IL2212 EMBEDDED SOFTWARE

Practical Homework 2

VERSION 1.1

Revision History

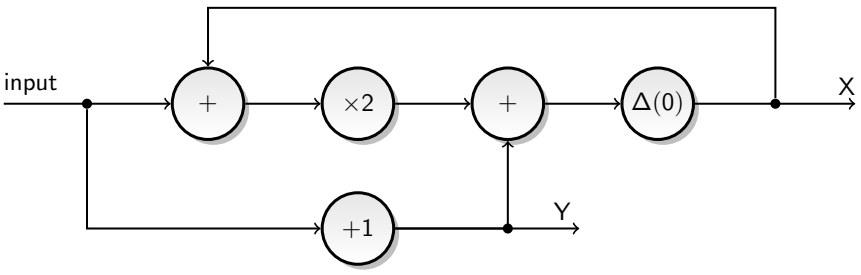
- Version 1.1: Correct version with 9 points (8 Point + 1 Bonus Point), task 4 and task 5 changed!
- Version 1.0: Initial version

Requirements

The total amount of points in this first homework is 8 points. To pass the practical homework, 17 points out of 24 points in all three homework examinations are required.

The total amount of points in this homework is 8 points and 1 bonus point. To pass the practical homework, 17 points out of 24 points in all three homework examinations are required.

1. (0.5 POINTS) Write a Lustre program that has two Boolean inputs `set` and `reset` and a Boolean output `y`, and implements the following function.
   - The output `y` should be set to `true`, if `set` has the value `true` and `reset` has the value `false`.
   - The output `y` should be set to `false`, if `set` has the value `false` and `reset` has the value `true`.
   - In all other cases, the output `y` should keep its previous value.
2. (0.5 POINTS) Write a Lustre program that implements the following application.



3. (1 POINTS) Write a Lustre program, which has a single input `x` and an output `y`, and shall implement the following function.

   - The output `y` shall be the sum of the current input `a` and the value of `a` at the previous (or current) even cycle. The program starts with cycle 1. The program shall be implemented using the temporal operators `when` and `current`.

     The following output illustrates the behaviour of the program.

```
#inputs "x":int
#outputs "y":int
#step 1
1
1
#step 2
2
4
#step 3
3
5
#step 4
4
8
#step 5
5
9
#step 6
6
12
```

4. (2 POINTS)

   (a) Write a Lustre program that has two Boolean inputs, `coin5` and `coin10`, and two Boolean outputs, `bottle` and `return_coin`, and implements the following control function for a machine that takes coins and returns a bottle, when the right amount (10 SEK) is inserted. If too much money is inserted, a bottle and a coin (5 SEK) shall be returned.

   - The input `coin5` is active, if a coin with value 5 SEK is inserted.
   - The input `coin10` is active, if a coin with value 10 SEK is inserted.
   - The inputs `coin5` and `coin10` cannot be active in the same time instance.
   - The output `bottle` shall be active, if at least 10 SEK are inserted. Then also the internal counter shall be reduced corresponding to 10 SEK.
   - The output `return_coin` shall be active, if 15 SEK have been inserted. Then the internal counter shall be reduced to 0 SEK, since also a bottle will be ejected at the same time instance.

   (b) Write a C-program that implements the corresponding functionality as the Lustre-program and reads the input values from the terminal and writes the output values to the terminal.

5. (4 POINTS and 1 BONUS POINT) Analyse the following ForSyDe synchronous data flow (SDF) image application model, which is available on

   - https://gits-15.sys.kth.se/ugeorge/il2212-lab/tree/master/model

   and complete the following tasks:

(a) Follow the explanations on the GitHub page of the application and run a few examples. No answer needs to given to this subtask.

(b) Create an own `ppm`-file of a picture with a modest size, we recommend less then 20.000 pixels, because of the exponentially increasing run-time, and apply the program on it.

In Linux the following command can be used to convert a `jpg`-file to `ppm`-format:

```
convert source.jpg -compress none -resize 120x80 image-dir/target.ppm
```

Be careful that you keep the correct `X:Y` relation of the original picture when downsizing. In other operating systems there exist corresponding commands or tools that can do the conversion. Give a screenshot, when executing the program and give the final image.

(c) Derive and draw the SDF graph of the SDF image application model.

(d) Give a short description of the actors in the SDF graph and an overview description of the function it executes.

(e) For each arc (signal), give the data type for the tokens, and an estimate of the token size in bits. Assume that integers require 32 bits, and a floating-point variable 64 bits. Also assume an image size of `X x Y` pixels.

(f) Give a formula using actor runtimes as parameters to derive the run-time of the application running on a single processor. You can neglect the time for communication.

(g) (BONUS) Sort the actors according to their expected run-time. Give a short motivation.

(h) (BONUS) Suggest possible ideas for a possible parallel implementation, taking the data dependencies into account.