CS684 Spring 2024-25

CS684: Embedded System Course

Assignment 2: Lustre/Heptagon

Note : Assignment 2 contains theory questions as well as coding questions. For theory questions, submit your answer in the text file and for coding questions, submit code files (.lus or .ept). Before submitting, refer to the Submission Instructions.

PROBLEM STATEMENTS

Download the file `roll#no-_assignment2.tar.xz`. Extract using command `tar -xzvf roll#no-_assignment2.tar.xz`
The skeleton files are already provided according to the given folder structure (mentioned at the end). For the questions that require code to be written, you should complete the node definitions already provided.
For questions that require text files, create a new text file, add the answer to it and name it according to the instructions provided

---

Q1.

Complete the following table giving values of expressions in column 1 for the first 6 cycles. The values of integer flows for the first 5 cycles are given in rows, labelled P and Q.

Expression\Cycle						
P						
Q						

pre(Q)

P + 1

P -> Q

P ->
(pre(Q)+1)

Q fby P

Answer format: Create `q1.txt` text file. List the values of the expressions for 6 cycles in comma-separated format. Make sure to write each expression's value for the six cycles on a separate line. For ex.

1,1,1,1,1,1

2,2,2,2,2,2

3,3,3,3,3,3

Q2.

Consider the following Lustre nodes:

```
node foo(P: int) returns (Q, R: int)
var W, Y, Z: int;
let
   Q  =  W* Z;
   W = 0 -> (pre(Z) +1);
   Z = W + Y;
   Y =  sqr(W);
```

```
    R = Q + W;
tel

node sqr(A :int) returns (B: int)
let
    B = A*A;
tel
```

Is this program causally correct? What is the order in which values of variables Q,R,W,Y,Z are calculated in each cycle?

Answer format: Create `q2.txt` text file. Write answers of both questions in comma separated format. For ex.

```
no
Q,R,W,Y,Z
```

Q3.

Write following lustre codes:

3.1: Define a Lustre node named **`progression`** for returning the following sequence of values. `(Code required)`

1, 9, 36, 100,...

(Hint: Consider how the difference of two successive terms grows.)
Note: In lustre, zero input nodes are not allowed. If your node is properly simulated in heptagon you can submit it. But keep the extension as .lus.

3.2: Complete the definition of following Lustre node `(Code required)`

```
node gen(req:bool) returns (ack: bool)
...
```

such that **ack is true** in the current cycle if and only if **req** has been:

- true for the current cycle and
- false for the last cycle and
- true for the second last cycle.

(Hint: First count for how many previous cycles req has been true continuously.)

Answer format: For question 3.1 & 3.2, submit .lus file. The skeleton file is already provided. Do not change the name of the node and the input and output prarameters.

Q4.

Answer the following questions on Heptagon nodes.

4.1: Please describe in English the output produced by the following Heptagon node.

```
node t(x: bool^5) returns (y: bool);
let
   y = fold<<5>>(or)(x, false)
tel
```

4.2: What happens if you change the equation to `y = fold<<5>>(or)(x, true)`? Describe the output.

4.3: Complete the definition of the following Heptagon node with parameter n. (Code required)

```
node mutex<<n>>(ack: bool^n) returns (ok: bool)
...
```

The node should check for mutual exclusion of ack[i] and ack[j]. That is 'ok' is true provided expression ack[i] and ack[j] is false for all i, j pairs with i not= j.

Answer format: Create `q41 & q42` text files. No specific format is required for these questions. These question will be manually evaluated. For q4.3, add your code to the skeleton file provided. Do not change the name of the node and the input and output prarameters.

Q5.

Study the following Heptagon code for ripple adder.

```
node mxor(x, y: bool) returns (c: bool)
let
   c = (x and not y) or ((not x) and y);
tel

node fa(x, y, cin: bool) returns (z, cout: bool)
let
   z = mxor(mxor(cin, x), y);
   cout = if cin then (x or y) else (x and y);
tel

node rippleadd<<n:int>>(a: bool^n; b: bool^n) returns (c: bool^n; over: bool)
let
   (c,over) = mapfold<<n>>fa(a, b, false);
tel
```

A sample code file with the above code simulating a 4-bit adder is given: Download File.

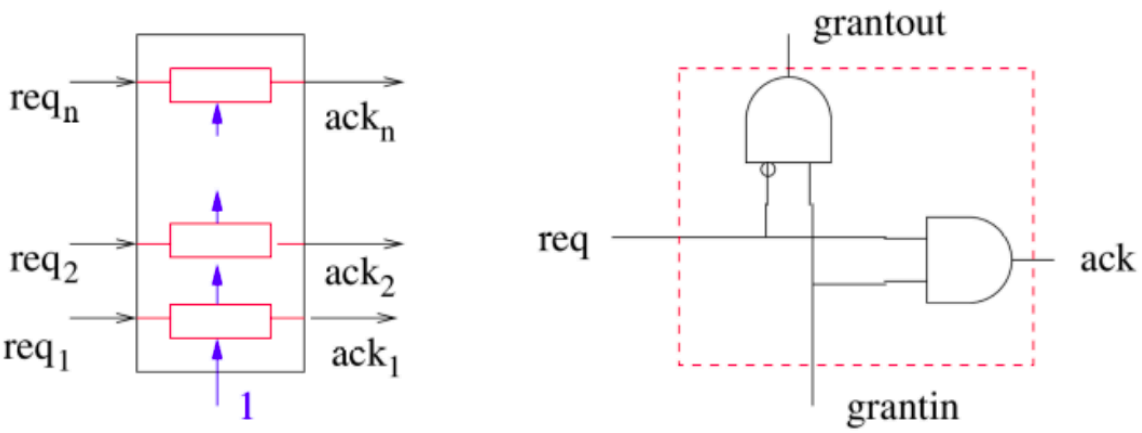
`5.1:` Manually compute the output at 0th cycle of rippleadd for constants a=[0,1,1,0,1,1,1,0] and b=[1,1,0,1,0,1,1,1] given as input to rippleadd<<8>>(a,b).

`5.2:` Using this node 'rippleadd', define a node counter which counts in binary modulo 64. (It should output unsigned 8 bit binary numbers corresponding to the decimal numbers 0, 1, 2, ..., 63, 0, 1, 2,.... in successive cycles). `(Code required)`

Answer format: For question 5.1, create `q51` text file. Write the answer of the question in comma separated format (write both values in a single line). For q5.2, add your code to the skeleton file provided. Do not change the name of the node and the input and output prarameters.

Q6.

6.1: **(Synchronous Bus Arbiter)** An arbiter arbitrates between multiple requests coming at each cycle and gives acknowledgement to at most one of them. Consider the arbiter circuit below. (Code required)



Each **red box** (denoting a cell) in the left hand side figure is expanded to the **cell** circuit given in the right hand side figure.

Using Heptagon, model each cell as a

> node cell(req, grantin: bool) returns (ack, grantout: bool)

Model a 5 cell arbiter **as an assembly of 5 cells** as shown in the figure. Call this

> node arbiter(req:bool^5) returns (ack:bool^5)

> Hint: See the implementation of rippleadd given in the previous question. Extra credit will be given if you can program this as an n cell arbiter with parameter n.

Define a suitable display node to show the output. Simulate the arbiter using the tool Heptagon and check its functioning. Submit a screenshot of sample output using the sim2chro display.

6.2: Which of the following properties does this arbiter have?

a. Mutual exclusion of ack[i], ack[j] for i not= j.
b. No spurious ack, i.e. ack[i] => req[i]
c. No lost cycles, i.e. in any cycle, if there is at least 1 true request, the arbiter should have at least one true ack.

Answer format: For q6.1, add your code to the skeleton file provided. Do not change the name of the node and the input and output prarameters. For question 6.2, create `q62` text file. Write the options in comma separated format.

Q7.

A monitor node for a property S takes as input a set of flows to observe (e.g. p, q: bool). It outputs a single boolean flow ok. The idea is that at every clock cycle, ok is true if the property S holds for the past sequence of inputs (including the current cycle). For example: Property S: “p is continuously true in the past” has the monitor node

```
node smonitor(p: bool) returns (ok: bool)
let
   ok = p -> (pre(ok) and p);
tel
```

Answer the following

`7.1`: Give the output of the above **smonitor** node for the input flow p = true true true false true false true true

`7.2`: Give a monitor for the following property: “p is continuously true in the past AND q has occured at least once in the past.” `(Code required)`

`7.3`: Give a monitor node for the following property: Assume that a, b, c are boolean flows. “Everytime a occurs, c will remain continuously true from then on until a b occurs”. Specify additional assumptions that you make in your design (E.g. what happens if a and b occur simultaneously). `(Code required)`

`7.4`: Give a monitor node for the following property: Assume that req and ack are boolean flows. “If req has been true for the last 3 cycles (including the current cycle) then ack must be true in the current cycle.” `(Code required)`

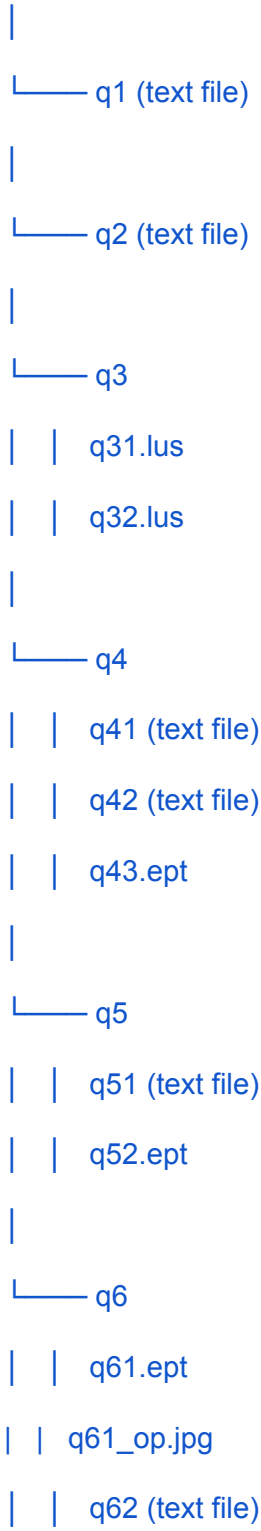
Answer format: For question 7.1, create `q71` text file. Write the answer the question in comma separated format. For q7.2, q7.3 and q7.4 add your code to the skeleton file provided. Do not change the name of the node and the input and output prarameters.

Clarification: req and ack are both input boolean flows. ok is the output flow. ok is true if the property is observed/followed.

Submission Instructions:

- Create a folder named **`<RollNo>_assignment2`**
- Copy and paste all the text files which has to be submitted inside the newly created folder according to the structure as shown below:

<RollNo>_assignment2



```
|
└── q7
    | q71 (text file)
    | q72.ept
    | q73.ept
    | q74.ept
```

Note : Folder name and file name should be same as mentioned in the above structure.

If you fail to follow the naming and directory structure, zero marks will be awarded

- Compress the folder in a `.tar.gz` file and submit it on moodle.
