**Custom Computing Assessed Coursework**

* 1. Rsh Definition:  
       
     rsh = <a,<b,z>> $wire <<a,b>,z>.  
       
     Rebecca code for P1:

INCLUDE "prelude.rby".

n = 4.

P1 = (Q1)^n ; (fork)^~1.

Q1 = snd fork ; rsh ; [add, (DI 0)^~1].

current = P1.

Diagram of P1:

Simulation:

re " x1 ; x2 ; x3 ; x4 ; x5 ; x6"

Simulation start :

0 - <x1,0> ~ ((((x1 + 0) + 0) + 0) + 0)

1 - <x2,0> ~ ((((x2 + 0) + 0) + 0) + ((((x1 + 0) + 0) + 0) + 0))

2 - <x3,0> ~ ((((x3 + 0) + 0) + ((((x1 + 0) + 0) + 0) + 0)) + ((((x2 + 0) + 0) + 0) + ((((x1 + 0) + 0) + 0) + 0))

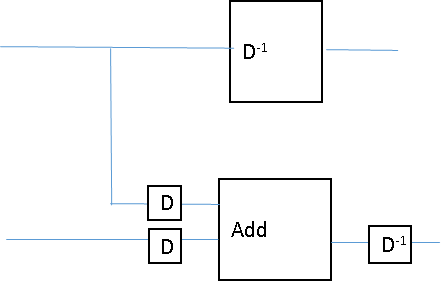
3 - <x4,0> ~ ((((x4 + 0) + ((((x1 + 0) + 0) + 0) + 0)) + ((((x2 + 0) + 0) + 0) + ((((x1 + 0) + 0) + 0) + 0))) + (

4 - <x5,((((x1 + 0) + 0) + 0) + 0)> ~ ((((x5 + ((((x1 + 0) + 0) + 0) + 0)) + ((((x2 + 0) + 0) + 0) + ((((x1 + 0) ) + ((((x4 + 0) + ((((x1 + 0) + 0) + 0) + 0)) + ((((x2 + 0) + 0) + 0) + ((((x1 + 0) + 0) + 0) + 0))) + ((((x3 + 0) +

5 - <x6,((((x2 + 0) + 0) + 0) + ((((x1 + 0) + 0) + 0) + 0))> ~ ((((x6 + ((((x2 + 0) + 0) + 0) + ((((x1 + 0) + 0) (((x4 + 0) + ((((x1 + 0) + 0) + 0) + 0)) + ((((x2 + 0) + 0) + 0) + ((((x1 + 0) + 0) + 0) + 0))) + ((((x3 + 0) + 0) + 0)) + ((((x2 + 0) + 0) + 0) + ((((x1 + 0) + 0) + 0) + 0))) + ((((x3 + 0) + 0) + ((((x1 + 0) + 0) + 0) + 0)) + ((((x2 (x1 + 0) + 0) + 0) + 0))) + ((((x3 + 0) + 0) + ((((x1 + 0) + 0) + 0) + 0)) + ((((x2 + 0) + 0) + 0) + ((((x1 + 0) + 0)

* 1. We begin by placing delays at the domain and antidelays at the range of the adders, making Q2, which is described by

snd fork ; rsh ; [[D,D] ; add ; D-1 , D-1]  
  
and looks like:



Using Horner’s Rule, we can transform ([D,D] ; add ; D-1)n into ([D,D] ; add)n ; D-n

3 a) failure: a deviation from a design specification

error: a failure that results in an incorrect signal value

fault: an error that manifests as an incorrect logical result

b)