CS580
Homework 08:
Zeros Elimination.
Yie Sheng Chen 1303

Introduction

A ring of synchronous processors are to be designed to circulate values and, at the same time, do garbage collection by turning off processor at the end of the pipeline when it holds two zeros at a particular execution step. Here zeros values are used to indicate garbage.

notations used:

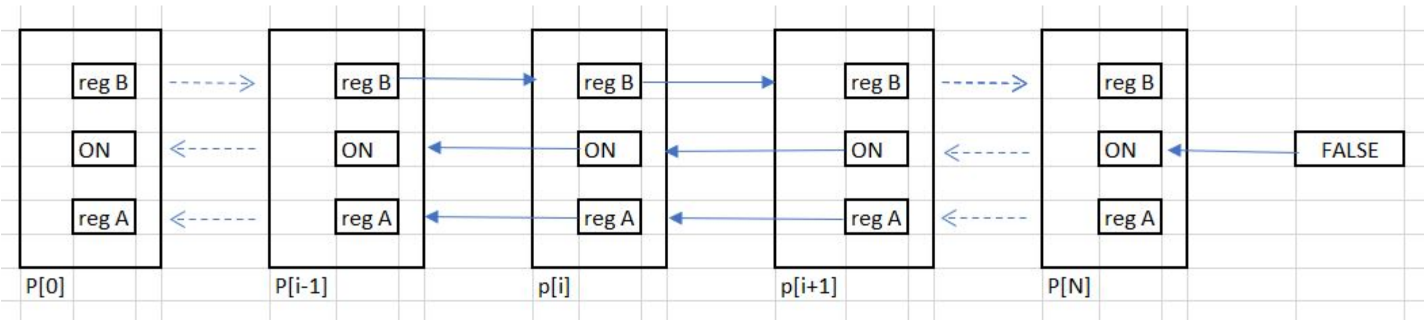
```
 |-> : leads to
=> : logical implication
<=> : if and only if
inv. q : q is invariant
{comments} : comments are enclosed in curly brackets
```

Processor Design

```
Explanation:

Read access is indicated by an arrow
from the register to the edge of the processor.

A fixed "False" value is hardwired to input of ON[N]
and this corresponds to a definition "ON(N+1)=false" in the always section.
```



Formal Specification

```
 init leads to post
stable post

Assume A0, B0 holds the original state of the A, B registers.

{function definition:}
count(X) : <+ i : 0 ≤i≤N :: 1 if ON[i] ^ X[i]=0>

{init:}
count(A) + count(B) = #zeros in A0 and B0
{post:}
count(A) + count(B) = 0 V count(A) + count(B) = 1
```

Programming Solution

```
{structure of the UNITY program is as the following:}
code-for-processor(0) ||
<|| i : 1≤i≤N : code-for-processor(i) >


Program ZerosElimination
declare
    A,B : array[0..N] of integer
    ON  : array[0..N] of Boolean
initially
    <[] i : 0≤i≤N :: ON[i] = True>
always
    isLast(i) = ¬ON[i+1] ^ ON[i] []
    ON(N+1) = False
assign
    B[0],A[0] := A[0],A[1] if i=0
    ||
    <|| i : 1≤i≤N ::
           B[i],A[i] := B[i-1],A[i+1]
                     if ON[i] ^ ¬isLast(i) ||
           B[i],A[i] := B[i-1],B[i]
                     if ON[i] ^ isLast(i) ^ ¬B[i]=0 ||
           A[i] := B[i-1]
                     if ON[i] ^ isLast(i) ^ B[i]=0 ^ ¬B[i-1]=0 ||
           ON[i],A[i] := False,B[i-1]
                     if ON[i] ^ isLast(i) ^ B[i]=0 ^ B[i-1]=0
           >
end
```

a sample scenario:

```
B[0~N] = 0 0 1 0     1 0 0 0     0 1 0 |0     1 0 0 |0     1 1 |0 0
A[0~N] = 1 0 1 1 --> 0 1 1 1 --> 1 1 1 |0 --> 1 1 1 |0 --> 1 1 |0 0
```

Formal Verification

1. stable post

```
Assume post "count(A) + count(B) = 0 V count(A) + count(B) = 1"
is true

Looking at the only statement in the UNITY program ZerosElimination,

count() only counts number of 0s in A or B registers of processors that are ON.

The values of A and B circulates and no new values are added.

And ON[i] only changes from True to False not False to True.

Since no new values is added to the ring
and once a processor is turned off,
there is no statement to turn it on,
the number of 0s in processors that are ON will remains 0 or 1.
```

2. init leads-to post

```
From "initially" section of the UNITY program,
"count(A) + count(B) = #zeros in A0 and B0" holds

because all processors are ON initially,
and the number of 0s in A or B registers of processors that
are ON equals the number of 0s in A0 and B0 which are
the initial state of A and B registers in the processors.

select the well-founded metric to be:
X = count(A) + count(B)

This metric is well-founded because its minimum value is 0

And X decreases which can be proved from
{X=k} leads to {X<k}

since by looking at the only statement in the UNITY program ZerosElimination,

before X=0 or X=1 holds,

if two 0s pass through the last processor that is ON,
the processor will be turned off and thus decreases X.
```