SoCV HW3 Report

鄭宇廷 B08202013

1. TODOs implementation

(1) buildNtkBdd: Perform DFS traversal from DFF inputs, inout and output gates. For each gate type, use for loop to get the corresponding CirGate (getRi(i)/getPi(i)/getPo(i)) and feed them into buildBdd function to build BDD for ntk.

(2) buildBdd: Since the input CirGate gate has perform DFS reorder, I use a for loop to traverse the reordered gate and check their type. If the type is AIG_GATE, I access the fanins of the gate, AND them and store the result BddNode. If the type is PO_GATE, I directly store the gate into the BddNode.

(3) buildPInitialState: This function is meant to set all state x0, x1, x2, … to zero. The expression is identical to (~x0)&(~x1)&(~x2)… = 1. Therefore, I set _initState as BddNodeV::_one initially and AND it with all the other state.

(4) buildPTransRelation: Since TR(Y,X,I) = (y0 = delta0) & (y1 = delta1) & ..etc. Similar to buildPInitialState, we can set _tri to BddNodeV::_one initially and AND it with all (y0 = delta0), (y1 = delta1), etc. Since = is identical to ~XOR logically, we can AND _tri with ~(y0 ^ delta0), ~(y1 ^ delta1), etc. to built _tri. After we build _tri, _tr can be simply obtained by _tr = exist I (_tri).

(5) buildPImate: According to the course slides p.25:
$S_{n+1}(Y)$ = exist x (TR(Y,X) & $S_n(X)$)
$S_{n+1}(X) = S_{n+1}(Y)|_{Y->X}$
$R_{n+1}(X) = R_n(X)$ & $S_{n+1}(Y)$
In the last step ,check if $R_{n+1}(X) = R_n(X)$ before update _reachStates. If $R_{n+1}(X) = R_n(X)$, set _isFixed = True.

(6) runPCheckProperty: First, create a BddNodeV check = monitor & _reachStates to see if the monitor is in the set of reachable states. If the BddNodeV check has paths to terminal BddNodeV::_one, it means that there’re counter example. Therefore, I propogate backward to find the input that will result in the counter example. Specifically, the output message shows the input that will result in that counter example. For instance, **“(1)” indicate that the first input has to be 1, and “!(1)”**

indicate that the first input has to be 0. Note that the index of input starts from 1.

2. Assertions

In the abstracted vending machine, I define three assertions below:

```
assign z0 = initialized && (serviceTypeOut == `SERVICE_ON) && (itemTypeOut != `ITEM_NONE);
assign z1 = initialized && (serviceTypeOut == `SERVICE_OFF) && (itemTypeOut == `ITEM_NONE) && (exchangeReady == 1'b0);
assign z2 = initialized && (serviceTypeOut == `SERVICE_ON) && (inputValue > 5'd30);
```

z0: the itemTypeOut should be `ITEM_NONE in `SERVICE_ON state

z1: When the exchange is not read yet, the itemTypeOut should not be `ITEM_NONE

z2: the input value is limited to 30 in `SERVICE_ON state

3. My Verification result

(1) a.dofile

```
setup> cirread ./design/SoCV/basic/a.v
Converted 0 1-valued FFs and 16 DC-valued FFs.

setup> breset 2000 8009 30011

setup> bsetorder -file
Set BDD Variable Order Succeed !!

setup> bconstruct -all

setup> set system vrf

vrf> pinit init

vrf> ptrans tri tr

vrf> usage
Period time used : 2.12 seconds
Total time used  : 2.12 seconds
Peak memory used   : 51.01 M Bytes
Total memory used  : 47.96 M Bytes
Current memory used: 86.46 M Bytes

vrf> pimage -n 120
Fixed point is reached (time : 12)

vrf> usage
Period time used : 1.36 seconds
Total time used  : 3.48 seconds
Peak memory used   : 51.26 M Bytes
Total memory used  : 48.21 M Bytes
Current memory used: 86.72 M Bytes

vrf> pcheckp -o 0
Monitor "z1[0]" is safe.

vrf> pcheckp -o 1
Monitor "z2[0]" is safe.

vrf> pcheckp -o 2
Monitor "z3[0]" is safe.

vrf> pcheckp -o 3
Monitor "z4[0]" is safe.

vrf> pcheckp -o 4
Monitor "z5[0]" is safe.

vrf> quit -f
```

(2) b.dofile

```
setup> cirread ./design/SoCV/basic/b.v
Converted 0 1-valued FFs and 10 DC-valued FFs.

setup> breset 2000 8009 30011

setup> bsetorder -file
Set BDD Variable Order Succeed !!

setup> bconstruct -all

setup> set system vrf

vrf> pinit init

vrf> ptrans tri tr

vrf> usage
Period time used : 0.1 seconds
Total time used  : 0.1 seconds
Peak memory used   : 22 M Bytes
Total memory used  : 18.84 M Bytes
Current memory used: 57.35 M Bytes

vrf> pimage -n 120
Fixed point is reached (time : 107)

vrf> usage
Period time used : 0.54 seconds
Total time used  : 0.64 seconds
Peak memory used   : 22.87 M Bytes
Total memory used  : 19.62 M Bytes
Current memory used: 58.12 M Bytes

vrf> pcheckp -o 0
Monitor "p1[0]" is safe.

vrf> pcheckp -o 1
Monitor "p2[0]" is safe.

vrf> pcheckp -o 2
Monitor "p3[0]" is safe.

vrf> pcheckp -o 3
Monitor "p4[0]" is safe.

vrf> pcheckp -o 4
Monitor "p5[0]" is violated.
Counter Example:
0: (1)
1: (1)
2: (1)
3: (1)
4: (1)
5: (1)
6: (1)
7: (1)
8: (1)
9: (1)
10: (1)
11: (1)
12: (1)
13: (1)
14: (1)
15: (1)
16: (1)
17: (1)
18: (1)
19: (1)
20: (1)
21: (1)
22: (1)
23: (1)
24: (1)
25: (1)
26: (1)
27: (1)
28: (1)
29: (1)
30: (1)
31: (1)
32: (1)
33: (1)
34: (1)
35: (1)
36: (1)
37: (1)
38: (1)
39: (1)
40: (1)
41: (1)

vrf> quit -f
```

Note: In counter example, “(1)” means the first input should be 1, which indicate that “11” and “10” are both valid counter examples.

(3) c.dofile

```
setup> cirread ./design/SoCV/basic/c.v
Converted 0 1-valued FFs and 6 DC-valued FFs.

setup> breset 2000 8009 30011

setup> bsetorder -file
Set BDD Variable Order Succeed !!

setup> bconstruct -all

setup> set system vrf

vrf> pinit init

vrf> ptrans tri tr

vrf> usage
Period time used : 0.07 seconds
Total time used  : 0.07 seconds
Peak memory used   : 21.02 M Bytes
Total memory used  : 18.03 M Bytes
Current memory used: 56.53 M Bytes

vrf> pimage -n 120
Fixed point is reached (time : 9)

vrf> usage
Period time used : 0.01 seconds
Total time used  : 0.08 seconds
Peak memory used   : 21.14 M Bytes
Total memory used  : 18.03 M Bytes
Current memory used: 56.53 M Bytes

vrf> pcheckp -o 0
Monitor "z0[0]" is violated.
Counter Example:
No counter example. Contradiction occur in initial state.

vrf> pcheckp -o 1
Monitor "z1[0]" is violated.
Counter Example:
0: !(1)
1: (1)
2: !(1)
3: (1)
4: !(1)

vrf> pcheckp -o 2
Monitor "z2[0]" is violated.
Counter Example:
0: input can be any random input
1: (1)
2: !(1)
3: (1)
4: !(1)
5: (1)

vrf> pcheckp -o 3
Monitor "z3[0]" is safe.

vrf> quit -f
```

Note: In counter example, “!(1)” means the first input should be 0, which indicate that “00” and “01” are both valid counter examples.

4. The Comparison with the ref program

(1) ptrans tri tr

	gv	gv-ref
a.dofile	vrf> ptrans tri tr vrf> usage Period time used : 2.05 seconds Total time used : 2.05 seconds Peak memory used : 51.13 M Bytes Total memory used : 47.96 M Bytes Current memory used: 86.46 M Bytes	vrf> ptrans tri tr vrf> usage Period time used : 0.76 seconds Total time used : 0.76 seconds Peak memory used : 50.22 M Bytes Total memory used : 33.21 M Bytes Current memory used: 71.65 M Bytes
b.dofile	vrf> ptrans tri tr vrf> usage Period time used : 0.1 seconds Total time used : 0.1 seconds Peak memory used : 22.25 M Bytes Total memory used : 18.84 M Bytes Current memory used: 57.35 M Bytes	vrf> ptrans tri tr vrf> usage Period time used : 0.02 seconds Total time used : 0.02 seconds Peak memory used : 21.22 M Bytes Total memory used : 4.094 M Bytes Current memory used: 42.53 M Bytes
c.dofile	vrf> ptrans tri tr vrf> usage Period time used : 0.07 seconds Total time used : 0.07 seconds Peak memory used : 21.03 M Bytes Total memory used : 18.03 M Bytes Current memory used: 56.53 M Bytes	vrf> ptrans tri tr vrf> usage Period time used : 0.01 seconds Total time used : 0.01 seconds Peak memory used : 20.36 M Bytes Total memory used : 3.281 M Bytes Current memory used: 41.72 M Bytes

(2) pimage

	gv	gv-ref
a.dofile	vrf> pimage -n 120 Fixed point is reached (time : 12) vrf> usage Period time used : 1.32 seconds Total time used : 3.37 seconds Peak memory used : 51.38 M Bytes Total memory used : 48.22 M Bytes Current memory used: 86.72 M Bytes	vrf> pimage -n 120 Fixed point is reached (time : 12) vrf> usage Period time used : 0.34 seconds Total time used : 1.1 seconds Peak memory used : 50.47 M Bytes Total memory used : 33.34 M Bytes Current memory used: 71.78 M Bytes
b.dofile	vrf> pimage -n 120 Fixed point is reached (time : 107) vrf> usage Period time used : 0.53 seconds Total time used : 0.63 seconds Peak memory used : 23 M Bytes Total memory used : 19.62 M Bytes Current memory used: 58.12 M Bytes	vrf> pimage -n 120 Fixed point is reached (time : 107) vrf> usage Period time used : 0.1 seconds Total time used : 0.12 seconds Peak memory used : 21.97 M Bytes Total memory used : 4.867 M Bytes Current memory used: 43.3 M Bytes
c.dofile	vrf> pimage -n 120 Fixed point is reached (time : 9) vrf> usage Period time used : 0.01 seconds Total time used : 0.08 seconds Peak memory used : 21.15 M Bytes Total memory used : 18.03 M Bytes Current memory used: 56.53 M Bytes	vrf> pimage -n 120 Fixed point is reached (time : 9) vrf> usage Period time used : 0.01 seconds Total time used : 0.02 seconds Peak memory used : 20.48 M Bytes Total memory used : 3.281 M Bytes Current memory used: 41.72 M Bytes

5. The advanced techniques and/or abstraction of the design

To abstract vending.v while maintaining the functionality of vending machine, I reduce the coin type to only NTD_10, item type to only ITEM_A and item cost to only COST_A. Also, I save some register bandwidth in order to reduce BDD size. After that abstraction, the verification result and its comparison with ref program are shown below:

	gv	gv-ref
Vending-abs.v	tim2811@venv2:~/socv/socv-1122$ ./gv setup> do ./hw/hw3/tests/vending-abs.dofile setup> cirread ./design/SoCV/vending/vending-abs.v Converted 0 1-valued FFs and 20 DC-valued FFs. setup> breset 2000 8009 30011 setup> bsetorder -file Set BDD Variable Order Succeed !! setup> bconstruct -all setup> set system vrf vrf> pinit init vrf> ptrans tri tr vrf> usage Period time used : 124.2 seconds Total time used : 124.2 seconds Peak memory used : 525.4 M Bytes Total memory used : 552.1 M Bytes Current memory used: 590.6 M Bytes vrf> pimage -n 120 Fixed point is reached (time : 11) vrf> usage Period time used : 53.72 seconds Total time used : 177.9 seconds Peak memory used : 526.9 M Bytes Total memory used : 552.1 M Bytes Current memory used: 590.6 M Bytes vrf> pcheckp -o 0 Monitor "z0[0]" is safe. vrf> pcheckp -o 1 Monitor "z1[0]" is safe. vrf> pcheckp -o 2 Monitor "z2[0]" is safe. vrf> quit -f	tim2811@venv2:~/socv/socv-1122$ ./hw/hw3/gv-ref setup> do ./hw/hw3/tests/vending-abs.dofile setup> cirread ./design/SoCV/vending/vending-abs.v Converted 0 1-valued FFs and 20 DC-valued FFs. setup> breset 2000 8009 30011 setup> bsetorder -file Set BDD Variable Order Succeed !! setup> bconstruct -all setup> set system vrf vrf> pinit init vrf> ptrans tri tr vrf> usage Period time used : 162.6 seconds Total time used : 162.6 seconds Peak memory used : 525.1 M Bytes Total memory used : 552.1 M Bytes Current memory used: 590.5 M Bytes vrf> pimage -n 120 Fixed point is reached (time : 11) vrf> usage Period time used : 35.15 seconds Total time used : 197.7 seconds Peak memory used : 526.8 M Bytes Total memory used : 552.1 M Bytes Current memory used: 590.5 M Bytes vrf> pcheckp -o 0 Monitor "z0[0]" is safe. vrf> pcheckp -o 1 Monitor "z1[0]" is safe. vrf> pcheckp -o 2 Monitor "z2[0]" is safe. vrf> quit -f

Therefore, all three assertions are safe.