Year/Semester: 2022-23/Spring

Course Code	Course Name	Credits	Tag	Grade	Credit/Audit
EE 691	R & D Project	6.0	Department elective	AA	C
EE 709	Testing and Verification of VLSI Circuits	6.0	Core course	BB	C
TD 640	Health and Wellness through a lifecycle approach	6.0	Institute elective	CC	C

Report Pro

Report: BDD Assignment (EE709)

Aditya Gudla

213076003

Analysis of a 4-bit adder

Consider a 4-bit adder without an incoming carry.

It has 8 inputs

x3 x2 x1 x0

y3 y2 y1 y0

and produces five output bits cout s3 s2 s1 s0

Q1. Consider the following subset A of the domain: those combinations of x and y such that an odd number of bits of x and an odd number of bits of y are 1. Using the BDD package, find the image of the set A in the range.

Solution : [C code - 213076003_Q1.c]

```
if var.8
 if var.9
  if var.10
   !var.12
  else if !var.10
   1
  endif var.10
 else if !var.9
  1
 endif var.9
else if !var.8
 if var.9
  1
 else if !var.9
  if var.10
   1
  else if !var.10
   if var.11
    1
   else if !var.11
    var.12
   endif var.11
  endif var.10
 endif var.9
endif var.8
```

Q2. Consider the following subset B of the range: the set of all 5-bit numbers c y3 y2 y1 y0 such that the number of bits in the number is odd. Using the BDD package, find the pre-image of the set B in the domain.

Solution : [C code - 213076003_Q2.c]

```
if var.0
  if var.1
    if var.2
      if var.3
        if var.4
          0: if var.5
            1: if var.6
              0
            else if !var.6
              var.7
            endif var.6
          else if !var.5
            var.6
          endif var.5
        else if !var.4
          2: if var.5
            !subformula 1
          else if !var.5
            subformula 1
          endif var.5
        endif var.4
      else if !var.3
        if var.4
          3: if var.5
            !var.7
          else if !var.5
            4: if var.6
              var.7
            else if !var.6
              !var.7
            endif var.6
          endif var.5
        else if !var.4
          5: if var.5
            var.7
          else if !var.5
            !var.7
          endif var.5
        endif var.4
      endif var.3
    else if !var.2
      if var.3
```

```
      if var.4
        6: if var.5
          7: if var.6
            var.7
          else if !var.6
            1
          endif var.6
        else if !var.5
          !var.6
        endif var.5
      else if !var.4
        8: if var.5
          !subformula 7
        else if !var.5
          subformula 7
        endif var.5
      endif var.4
    else if !var.3
      if var.4
        !subformula 4
      else if !var.4
        9: if var.5
          subformula 4
        else if !var.5
          !subformula 4
        endif var.5
      endif var.4
    endif var.3
  endif var.2
else if !var.1
  if var.2
    if var.3
      if var.4
        10: if var.5
          var.6
        else if !var.5
          !subformula 1
        endif var.5
      else if !var.4
        !subformula 2
      endif var.4
    else if !var.3
      if var.4
        11: if var.5
          subformula 4
        else if !var.5
          var.7
        endif var.5
```

```
      else if !var.4
        !subformula 5
      endif var.4
    endif var.3
  else if !var.2
    if var.3
      if var.4
        12: if var.5
          !var.6
        else if !var.5
          !subformula 7
        endif var.5
      else if !var.4
        !subformula 8
      endif var.4
    else if !var.3
      !subformula 9
    endif var.3
  endif var.2
endif var.1
else if !var.0
  if var.1
    if var.2
      if var.3
        if var.4
          subformula 2
        else if !var.4
          !subformula 0
        endif var.4
      else if !var.3
        if var.4
          subformula 5
        else if !var.4
          !subformula 3
        endif var.4
      endif var.3
    else if !var.2
      if var.3
        if var.4
          subformula 8
        else if !var.4
          !subformula 6
        endif var.4
      else if !var.3
        if var.4
          subformula 9
        else if !var.4
          subformula 4
```

```
      endif var.4
    endif var.3
  endif var.2
 else if !var.1
  if var.2
    if var.3
      if var.4
        !subformula 2
      else if !var.4
        !subformula 10
      endif var.4
    else if !var.3
      if var.4
        !subformula 5
      else if !var.4
        !subformula 11
      endif var.4
    endif var.3
  else if !var.2
    if var.3
      if var.4
        !subformula 8
      else if !var.4
        !subformula 12
      endif var.4
    else if !var.3
      if var.4
        !subformula 9
      else if !var.4
        subformula 9
      endif var.4
    endif var.3
  endif var.2
 endif var.1
endif var.0
```

Q3. Lets prove a property about a four bit adder: show (using BDD's) that every even 4-bit number can be expressed as a sum of two prime numbers.

Solution : [C code - 213076003_Q2.c]

```
if var.8
 if var.9
  if var.10
   0
  else if !var.10
   !var.12
```

```
    endif var.10
  else if !var.9
    !var.12
  endif var.9
else if !var.8
  if var.9
    if var.10
      1
    else if !var.10
      if var.11
        !var.12
      else if !var.11
        1
      endif var.11
    endif var.10
  else if !var.9
    if var.10
      1
    else if !var.10
      var.11
    endif var.10
  endif var.9
endif var.8
```

----Subset of subset A(even outputs when inputs are prime numbers)----

----------BDD name : e----------

```
if var.8
  if var.9
    if var.10
      0
    else if !var.10
      !var.12
    endif var.10
  else if !var.9
    !var.12
  endif var.9
else if !var.8
  if var.9
    !var.12
  else if !var.9
    if var.10
      !var.12
    else if !var.10
      if var.11
```

```
      !var.12
    else if !var.11
      0
    endif var.11
  endif var.10
 endif var.9
endif var.8
```


----------All possible nonzero even output numbers----------
----------BDD name : i----------

```
if var.8
 !var.12
else if !var.8
 if var.9
  !var.12
 else if !var.9
  if var.10
   !var.12
  else if !var.10
   if var.11
    !var.12
   else if !var.11
    0
   endif var.11
  endif var.10
 endif var.9
endif var.8
```


To prove: Every even 4-bit number can be expressed as a sum of two prime numbers
We need to prove: bdd 'e' is a subset of bdd 'i' i.e. ~e+i = 1 OR ~i.e = 0
----------Proof 1 : ~i.e = 0----------

0
Result is a zero BDD hence ~i.e = 0

----------Proof 2 : ~e+i = 1----------

1
Result is a one BDD hence ~e+i = 1

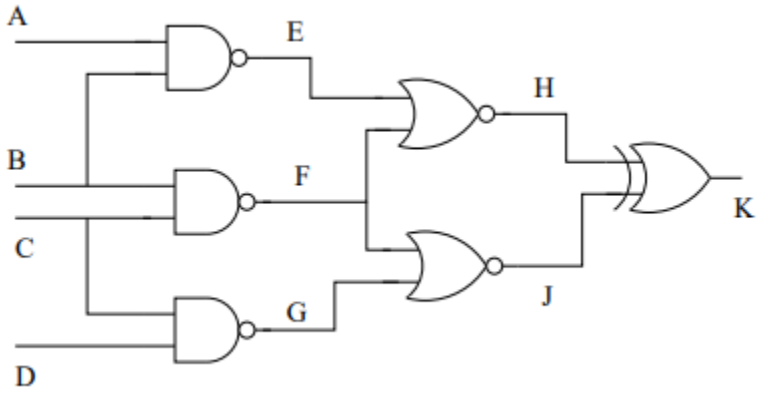
Hence the property is proved

Report: SAT Assignment (EE709)

Aditya Gudla

213076003

Q1.



a. Calculate a product of sums formula (CNF) which describes this network.
Solution:
(~A+~B+~E)(A+E)(B+E)(~B+~C+~F)(B+F)(C+F)(~C+~D+~G)(C+G)(D+G)(E+F+H)(~E+~H)(~F+~H)(F+G+J)(~F+~J)(~G+~J)(H+J+~K)(H+~J+K)(~H+J+K)(~H+~J+~K)

b. Using the minisat solver, find an input assignment such that the output K is 0.
Solution:

c nvars nclauses
p cnf 10 20
c index of the variables >> [A,B,C,D,E,F,G,H,J,K] = [1,2,3,4,5,6,7,8,9,10]
c clause 1 : ~A + ~B + ~E
-1 -2 -5 0
c clause 2 : A + E
1 5 0
c clause 3 : B + E
2 5 0
c clause 4 : ~B + ~C + ~F
-2 -3 -6 0
c clause 5 : B + F
2 6 0
c clause 6 : C + F
3 6 0
c clause 7 : ~C + ~D + ~G
-3 -4 -7 0
c clause 8 : C + G
3 7 0

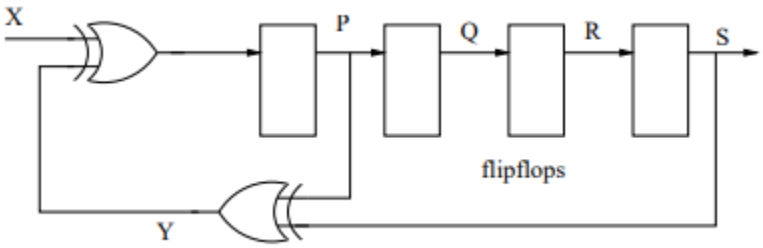
c clause 9 : D + G
4 7 0
c clause 10 : E + F + H
5 6 8 0
c clause 11 : ~E + ~H
-5 -8 0
c clause 12 : ~F + ~H
-6 -8 0
c clause 13 : F + G + J
6 7 9 0
c clause 14 : ~F + ~J
-6 -9 0
c clause 15 : ~G + ~J
-7 -9 0
c clause 16 : H + J + ~K
8 9 -10 0
c clause 17 : H + ~J + K
8 -9 10 0
c clause 18 : ~H + J + K
-8 9 10 0
c clause 19 : ~H + ~J + ~K
-8 -9 -10 0
c clause 20 : ~K
-10 0

c. Using the minisat solver, find an input assignment such that the output K is 1.
Solution:

c nvars nclauses
p cnf 10 20
c index of the variables >> [A,B,C,D,E,F,G,H,J,K] = [1,2,3,4,5,6,7,8,9,10]
c clause 1 : ~A + ~B + ~E
-1 -2 -5 0
c clause 2 : A + E
1 5 0
c clause 3 : B + E
2 5 0
c clause 4 : ~B + ~C + ~F
-2 -3 -6 0
c clause 5 : B + F
2 6 0
c clause 6 : C + F
3 6 0
c clause 7 : ~C + ~D + ~G
-3 -4 -7 0
c clause 8 : C + G
3 7 0
c clause 9 : D + G

4 7 0
c clause 10 : E + F + H
5 6 8 0
c clause 11 : ~E + ~H
-5 -8 0
c clause 12 : ~F + ~H
-6 -8 0
c clause 13 : F + G + J
6 7 9 0
c clause 14 : ~F + ~J
-6 -9 0
c clause 15 : ~G + ~J
-7 -9 0
c clause 16 : H + J + ~K
8 9 -10 0
c clause 17 : H + ~J + K
8 -9 10 0
c clause 18 : ~H + J + K
-8 9 10 0
c clause 19 : ~H + ~J + ~K
-8 -9 -10 0
c clause 20 : ~K
10 0

Q2.



a. Suppose the machine starts in the state P = Q = R = S = 1. Using minisat, find a sequence of input values at X which will take the machine to the state P = Q = R = S = 0.
Solution:

[Reaches after 4 cycles]

c nvars nclauses
p cnf 24 64
c index of the variables >>
[p0,q0,r0,s0,x0,p1,q1,r1,s1,x1,p2,q2,r2,s2,x2,p3,q3,r3,s3,x3,p4,q4,r4,s4]
c [1, 2, 3, 4, 5, 6, 7, 8, 9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24]
c Initial state
c clause1 IS0 : p0
1 0
c clause2 IS1 : q0

2 0
c clause3 IS2 : r0
3 0
c clause4 IS3 : s0
4 0
c Final state
c clause5 FS0 : ~p4
-21 0
c clause6 FS1 : ~q4
-22 0
c clause7 FS2 : ~r4
-23 0
c clause8 FS3 : ~s4
-24 0
c----------N = 1----------
c clause9 : ~p1 + ~x0 + ~p0 + s0
-6 -5 -1 4 0
c clause10 : ~p1 + ~x0 + p0 + ~s0
-6 -5 1 -4 0
c clause11 : ~p1 + x0 + ~p0 + ~s0
-6 5 -1 -4 0
c clause12 : ~p1 + x0 + p0 + s0
-6 5 1 4 0
c clause13 : p1 + ~x0 + ~p0 + ~s0
6 -5 -1 -4 0
c clause14 : p1 + ~x0 + p0 + s0
6 -5 1 4 0
c clause15 : p1 + x0 + ~p0 + s0
6 5 -1 4 0
c clause16 : p1 + x0 + p0 + ~s0
6 5 1 -4 0
c clause17 : ~p0 + q1
-1 7 0
c clause18 : p0 + ~q1
1 -7 0
c clause19 : ~q0 + r1
-2 8 0
c clause20 : q0 + ~r1
2 -8 0
c clause21 : ~r0 + s1
-3 9 0
c clause22 : r0 + ~s1
3 -9 0
c----------N = 2----------
c clause23 : ~p2 + ~x1 + ~p1 + s1
-11 -10 -6 9 0
c clause24 : ~p2 + ~x1 + p1 + ~s1
-11 -10 6 -9 0

c clause25 : ~p2 + x1 + ~p1 + ~s1
-11 10 -6 -9 0
c clause26 : ~p2 + x1 + p1 + s1
-11 10 6 9 0
c clause27 : p2 + ~x1 + ~p1 + ~s1
11 -10 -6 -9 0
c clause28 : p2 + ~x1 + p1 + s1
11 -10 6 9 0
c clause29 : p2 + x1 + ~p1 + s1
11 10 -6 9 0
c clause30 : p2 + x1 + p1 + ~s1
11 10 6 -9 0
c clause31 : ~p1 + q2
-6 12 0
c clause32 : p1 + ~q2
6 -12 0
c clause33 : ~q1 + r2
-7 13 0
c clause34 : q1 + ~r2
7 -13 0
c clause35 : ~r1 + s2
-8 14 0
c clause36 : r1 + ~s2
8 -14 0
c----------N = 3----------
c clause37 : ~p3 + ~x2 + ~p2 + s2
-16 -15 -11 14 0
c clause38 : ~p3 + ~x2 + p2 + ~s2
-16 -15 11 -14 0
c clause39 : ~p3 + x2 + ~p2 + ~s2
-16 15 -11 -14 0
c clause40 : ~p3 + x2 + p2 + s2
-16 15 11 14 0
c clause41 : p3 + ~x2 + ~p2 + ~s2
16 -15 -11 -14 0
c clause42 : p3 + ~x2 + p2 + s2
16 -15 11 14 0
c clause43 : p3 + x2 + ~p2 + s2
16 15 -11 14 0
c clause44 : p3 + x2 + p2 + ~s2
16 15 11 -14 0
c clause45 : ~p2 + q3
-11 17 0
c clause46 : p2 + ~q3
11 -17 0
c clause47 : ~q2 + r3
-12 18 0
c clause48 : q2 + ~r3

12 -18 0
c clause48 : ~r2 + s3
-13 19 0
c clause50 : r2 + ~s3
13 -19 0
c----------N = 3----------
c clause51 : ~p4 + ~x3 + ~p3 + s3
-21 -20 -16 19 0
c clause52 : ~p4 + ~x3 + p3 + ~s3
-21 -20 16 -19 0
c clause53 : ~p4 + x3 + ~p3 + ~s3
-21 20 -16 -19 0
c clause54 : ~p4 + x3 + p3 + s3
-21 20 16 19 0
c clause55 : p4 + ~x3 + ~p3 + ~s3
21 -20 -16 -19 0
c clause56 : p4 + ~x3 + p3 + s3
21 -20 16 19 0
c clause57 : p4 + x3 + ~p3 + s3
21 20 -16 19 0
c clause58 : p4 + x3 + p3 + ~s3
21 20 16 -19 0
c clause59 : ~p3 + q4
-16 22 0
c clause60 : p3 + ~q4
16 -22 0
c clause61 : ~q3 + r4
-17 23 0
c clause62 : q3 + ~r4
17 -23 0
c clause63 : ~r3 + s4
-18 24 0
c clause64 : r3 + ~s4
18 -24 0