Homework 7

Gabrielle Guarino

Problem 1:

Serial adder

Inputs: X1, X2, X3

Output: Z

X1, X2 and X3 are N bits long

Z could be more than N bits

1. Mealy state diagram  
   \*attached\*
2. Moore state diagram  
   \*attached\*
3. State assignments tried:  
     
   1: carry0, carry1, carry2   
   - this corresponds to the value being carried over  
   - states transition on x values  
     
   2: c00, c01, c10, c11   
   - this corresponds to the pending and currently waiting carried values  
   - states transition on x values  
     
   3: zero, one, two, three, four, five  
   - this corresponds to the values of the previous z and the previous carry out  
   - states transition on x values  
     
   Set of logic equations generated by espresso:

Problem 2:

|  |  |  |
| --- | --- | --- |
|  | I1 | I2 |
| A | -,- | F,0 |
| B | B,0 | C,0 |
| C | E,0 | A,1 |
| D | B,0 | D,0 |
| E | F,1 | D,0 |
| F | A,0 | -,- |

1. Find all maximum compatibles  
     
   Two states are compatible if after applying any input sequence to S1 and S2, the resulting output sequences are the same. Also the next states need to be compatible.

Compatible pairs:  
AB, AD, AE, AF, BF, CF, DF

Incompatible pairs:  
AC, BC, BD, BE, CD, CE, DE, EF

Constraint equation:  
X = (A’ + C’)(B’+ C’)(B’ + D’)(B’ + E’)(C’ + D’)(C’ + E’)(D’ + E’)(E’ + F’)  
X = (A’B’ + C’B’ + C’ + A’C’)(B’ + D’)(B’ + E’)(C’ + D’)(C’ + E’)(D’ + E’)(E’ + F’)  
X = (A’B’ + C’)(B’ + D’)(B’ + E’)(C’ + D’)(C’ + E’)(D’ + E’)(E’ + F’)  
X = (A’B’ + C’B’ + A’B’D’ + C’D’)(B’ + E’)(C’ + D’)(C’ + E’)(D’ + E’)(E’ + F’)  
X = (A’B’ + C’B’ + C’D’)(B’ + E’)(C’ + D’)(C’ + E’)(D’ + E’)(E’ + F’)  
X = (A’B’ + C’B’ + C’D’E’)(C’ + D’)(C’ + E’)(D’ + E’)(E’ + F’)  
X = (A’B’C’ + C’B’ + C’D’E’ + A’B’D’)(C’ + E’)(D’ + E’)(E’ + F’)  
X = (A’B’C’ + C’B’ + C’D’E’ + A’B’D’E’)(D’ + E’)(E’ + F’)  
X = (B’C’D’ + C’D’E’ + A’B’D’E’ + B’C’E’)(E’ + F’)  
X = (C’D’E’ + A’B’D’E’ + B’C’E’ + B’C’D’F’)  
  
MCs:  
ABF, CF, ADF, AE

1. Derive all prime compatibles

|  |  |  |
| --- | --- | --- |
| Ci | Implied States | ICs |
| ABF | AB, CF | CF |
| CF | AE | AE |
| ADF | AB, DF | AB |
| AE | DF | DF |
| AF |  |  |
| AB | CF | CF |
| BF | AB | AB |
| AD | DF | DF |
| DF | AB | AB |
| A |  |  |
| B |  |  |
| C |  |  |
| D |  |  |
| E |  |  |
| F |  |  |

‘

After removing the dominated Ci’s… (A, F)

Prime compatibles = ABF, CF, ADF, AE, AF, AB, BF, AD, DF, B, C, D, E

1. Construct a matrix formulation of the state covering problem as a binate covering problem

Cover table:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| PC | A | B | C | D | E | F | AB | AE | CF | DF |
| ABF | x | x |  |  |  | x | x |  | o |  |
| CF |  |  | x |  |  | x |  | o | x |  |
| ADF | x |  |  | x |  | x | o |  |  | x |
| AE | x |  |  |  | x |  |  | x |  | o |
| AF | x |  |  |  |  | x |  |  |  |  |
| AB | x | x |  |  |  |  | x |  | o |  |
| BF |  | x |  |  |  | x | o |  |  |  |
| AD | x |  |  | x |  |  |  |  |  | o |
| DF |  |  |  | x |  | x | o |  |  | x |
| B |  | x |  |  |  |  |  |  |  |  |
| C |  |  | x |  |  |  |  |  |  |  |
| D |  |  |  | x |  |  |  |  |  |  |
| E |  |  |  |  | x |  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| I | PCs | ICs |
| 1 | ABF | CF |
| 2 | ADF | AB |
| 3 | CF | AE |
| 4 | AE | DF |
| 5 | AF |  |
| 6 | AB | CF |
| 7 | BF | AB |
| 8 | AD | DF |
| 9 | DF | AB |
| 10 | B |  |
| 11 | C |  |
| 12 | D |  |
| 13 | E |  |

State coverage constraint terms:

If each c1, c2, and c3 covers some state A, we need the term (c1 + c2 + c3)

(C1 + C2 + C4 + C5 + C6 + C8)

(C1 + C6 + C7 + C10)

(C3 + C11)

(C2 + C8 + C9 + C12)

(C4 + C13)

(C1 + C2 + C3 + C5 + C7 + C9)

Implication constraint terms:

If c1 implies c2, and c2 is the only cover of c2, we need the term c1’ + c2

If c1 implies c2 and c2 is covered by c2 or c3, we need the terms c1’ + c2 + c3

(C1’ + C3)

(C2’ + C1 + C6)

(C3’ + C4)

(C4’ + C9 + C2)

(C6’ + C3)

(C7’ + C1 + C6)

(C8’ + C9 + C2)

(C9’ + C1 + C6)

Constraint equation:

(C1 + C2 + C4 + C5 + C6 + C8)(C1 + C6 + C7 + C10)(C3 + C11)

(C2 + C8 + C9 + C12)(C4 + C13)(C1 + C2 + C3 + C5 + C7 + C9)(C1’ + C3)

(C2’ + C1 + C6)(C3’ + C4)(C4’ + C9 + C2)(C6’ + C3)(C7’ + C1 + C6)(C8’ + C9 + C2)

(C9’ + C1 + C6) = 1

1. Solve the binate covering problem using any systematic approach. Create a state table for an FSM that is compatible with M1 but has the minimum number of states

(C1 + C2 + C4 + C5 + C6 + C8)(C1 + C6 + C7 + C10)(C3 + C11)

(C2 + C8 + C9 + C12)(C4 + C13)(C1 + C2 + C3 + C5 + C7 + C9)(C1’ + C3)

(C2’ + C1 + C6)(C3’ + C4)(C4’ + C9 + C2)(C6’ + C3)(C7’ + C1 + C6)(C8’ + C9 + C2)

(C9’ + C1 + C6) = 1

Assuming C1 = 1…

(C3 + C11)(C2 + C8 + C9 + C12)(C4 + C13)(C3)(C3’ + C4)(C4’ + C9 + C2)(C6’ + C3)(C8’ + C9 + C2) = 1

Assuming C1 = 0…

(C2 + C4 + C5 + C6 + C8)(C6 + C7 + C10)(C3 + C11)

(C2 + C8 + C9 + C12)(C4 + C13)(C2 + C3 + C5 + C7 + C9)

(C2’ + C6)(C3’ + C4)(C4’ + C9 + C2)(C6’ + C3)(C7’ + C6)(C8’ + C9 + C2)

(C9’ + C6) = 1

Assuming C1 = 1, C3 = 1…

(C2 + C8 + C9 + C12)(C4 + C13)(C4)(C4’ + C9 + C2)(C6’)(C8’ + C9 + C2) = 1

~~Assuming C1 = 1, C3 = 0…~~

~~(C11)(C2 + C8 + C9 + C12)(C4 + C13)(0)(C4’ + C9 + C2)(C6’)(C8’ + C9 + C2) = 0~~

~~Doesn’t work.~~

Assuming C1 = 0, C3 = 1…

(C2 + C4 + C5 + C6 + C8)(C6 + C7 + C10)

(C2 + C8 + C9 + C12)(C4 + C13)(C2 + C5 + C7 + C9)

(C2’ + C6)(C4)(C4’ + C9 + C2)(C6’)(C7’ + C6)(C8’ + C9 + C2)

(C9’ + C6) = 1

Assuming C1 = 1, C3 = 1, C4 = 1…

(C2 + C8 + C9 + C12)( C9 + C2)(C6’)(C8’ + C9 + C2) = 1

~~Assuming C1 = 1, C3 = 1, C4 = 0…~~

~~(C2 + C8 + C9 + C12)(C13)(0)(C6’)(C8’ + C9 + C2) = 0~~

~~Doesn’t work.~~

Assuming C1 = 0, C3 = 1, C4 = 1…

(C6 + C7 + C10)(C2 + C8 + C9 + C12)(C2 + C5 + C7 + C9)

(C2’ + C6)(C9 + C2)(C6’)(C7’ + C6)(C8’ + C9 + C2)

(C9’ + C6) = 1

~~Assuming C1 = 0, C3 = 1, C4 = 0…~~

~~(C2 + C5 + C6 + C8)(C6 + C7 + C10)(C2 + C8 + C9 + C12)(C13)(C2 + C5 + C7 + C9)~~

~~(C2’ + C6)(0)(C6’)(C7’ + C6)(C8’ + C9 + C2)(C9’ + C6) = 0~~

~~Doesn’t work.~~

Assuming C1 = 1, C3 = 1, C4 = 1, C9 = 1…

(C6’) = 1

C6 has to be 0 in this case.

**Solution = C1, C3, C4, C9**

Assuming C1 = 1, C3 = 1, C4 = 1, C9 = 0…

(C2 + C8 + C12)(C2)(C6’)(C8’ + C2) = 1

~~Assuming C1 = 0, C3 = 1, C4 = 1, C9 = 1…~~

~~(C6 + C7 + C10)(C2’ + C6)(C6’)(C7’ + C6)(C6) = 1~~

~~This is always 0 because it is ANDing C6’ and C6.~~

~~Doesn’t work.~~

Assuming C1 = 0, C3 = 1, C4 = 1, C9 = 0…

(C6 + C7 + C10)(C2 + C8 + C12)(C2 + C5 + C7)

(C2’ + C6)(C2)(C6’)(C7’ + C6)(C8’ + C2) = 1

Assuming C1 = 1, C3 = 1, C4 = 1, C9 = 0, C6 = 0…

(C2 + C8 + C12)(C2)(C8’ + C2) = 1

~~Assuming C1 = 1, C3 = 1, C4 = 1, C9 = 0, C6 = 1…~~

~~(C2 + C8 + C12)(C2)(0)(C8’ + C2) = 0~~

~~Doesn’t work.~~

~~Assuming C1 = 0, C3 = 1, C4 = 1, C9 = 0, C6 = 0…~~

~~(C7 + C10)(C2 + C8 + C12)(C2 + C5 + C7)~~

~~(C2’)(C2)(C7’)(C8’ + C2) = 1~~

~~This has both a C2 and C2’ ANDed together.~~

~~Doesn’t work.~~

~~Assuming C1 = 0, C3 = 1, C4 = 1, C9 = 0, C6 = 1…~~

~~(C2 + C8 + C12)(C2 + C5 + C7)(C2)(0)(C8’ + C2) = 0~~

~~Doesn’t work.~~

Assuming C1 = 1, C3 = 1, C4 = 1, C9 = 0, C6 = 0, C2 = 1…

1 = 1

**Solution = C1, C3, C4, C2**

~~Assuming C1 = 1, C3 = 1, C4 = 1, C9 = 0, C6 = 0, C2 = 0…~~

~~(C8 + C12)(0)(C8’) = 0~~

~~Doesn’t work.~~

Two solutions:

C1, C3, C4, C9 or

C1, C2, C3, C4

i.e.

ABF, CF, AE, DF

ABF, ADF, CF, AE

Same number of states.

I chose the first one.

ABF -> a

CF -> b

AE -> c

DF -> d

|  |  |  |
| --- | --- | --- |
|  | I1 | I2 |
| a | a,0 | b,0 |
| b | c,0 | a,1 |
| c | a,1 | d,0 |
| d | a,0 | d,0 |

Problem 3:

1. Identify all the equivalence classes among the states

State table:

|  |  |  |
| --- | --- | --- |
|  | 0 | 1 |
| S0 | S4, 0 | S2, 0 |
| S1 | S2, 0 | S0, 0 |
| S2 | S1, 0 | S6, 0 |
| S3 | S6, 0 | S0, 0 |
| S4 | S5, 1 | S1, 0 |
| S5 | S4, 0 | S3, 0 |
| S6 | S3, 0 | S6, 0 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| S1 | (~~S4, S2)~~  ~~(S2, S0)~~ |  |  |  |  |  |
| S2 | (~~S4, S1) (S6, S2)~~ | (S2, S1)  (S0, S6) |  |  |  |  |
| S3 | ~~(S6, S4)~~  ~~(S2, S0)~~ | (S2, S1)  (S0, S0) | (S1, S6)  (S6, S0) |  |  |  |
| S4 | X | X | X | X |  |  |
| S5 | (S3, S2)  (S4, S4) | (~~S2, S4)~~  ~~(S0, S3)~~ | ~~(S4,S1)~~  ~~(S3,S6~~ | ~~(S4,S6)~~  ~~(S3,S0)~~ | X |  |
| S6 | (~~S3, S4)~~  ~~(S6, S2)~~ | (S3, S2)  (S6, S0) | (S3, S1)  (S6, S6) | (S3, S6)  (S6, S0) | X | ~~(S3, S4)~~  ~~(S6, S3)~~ |
|  | S0 | S1 | S2 | S3 | S4 | S5 |

(S0, S5)

(S1, S2)

(S1, S3)

(S1, S6)

(S2, S6)

(S2, S3)

(S3, S6)

S0 = S5

S1 = S2 = S3 = S6

S4

1. Construct a state table for a minimum-state FSM that is equivalent to the FSM given

|  |  |  |
| --- | --- | --- |
|  | 0 | 1 |
| S0 | S4, 0 | S1, 0 |
| S1 | S1, 0 | S0, 0 |
| S4 | S0, 1 | S1, 0 |

1. State minimization using sis

Problem 4:

1. Use list-scheduling method to find a minimum-latency schedule for G.  
   Repeat for each resource:  
   - Determine list of ready operations  
   - Determine list of unfinished operations  
   - Select from ready such that time(selected) + time(unfinished) < resource constraint  
   - Schedule the selected members into the current step

L = 1 U1,1 = {v2} T1,1 = {} S1,1 = {v2}

U1,2 = {v1} T1,2 = {} S1,2 = {v1}

L = 2 U2,1 = {} T2,1 = {v2} S2,1 = {}

U2,2 = {} T2,2 = {} S2,2 = {}

L = 3 U3,1 = {v3} T3,1 = {} S3,1 = {v3}

U3,2 = {v4} T3,2 = {} S3,2 = {v4}

L = 4 U4,1 = {} T4,1 = {v3} S4,1 = {}

U4,2 = {} T4,2 = {} S4,2 = {}

L = 5 U5,1 = {} T5,1 = {} S5,1 = {}

U5,2 = {v5} T5,2 = {} S5,2 = {v5}

L = 6 U6,1 = {v6} T6,1 = {} S6,1 = {v6}

U6,2 = {} T6,2 = {} S6,2 = {}

L = 7 U7,1 = {} T7,1 = {v6} S7,1 = {}

U7,2 = {} T7,2 = {} S7,2 = {}

L = 8 U8,1 = {} T8,1 = {v6} S8,1 = {}

U8,2 = {} T8,2 = {} S8,2 = {}

L = 9 U9,1 = {v7} T9,1 = {} S9,1 = {v7}

U9,2 = {} T9,2 = {} S9,2 = {}

L = 10 U10,1 = {} T10,1 = {v7} S10,1 = {}

U10,2 = {} T10,2 = {} S10,2 = {}

L = 11 U11,1 = {} T11,1 = {} S11,1 = {}

U11,2 = {v8} T11,2 = {} S11,2 = {v8}

11 clock cycles

DFG:

\*attached\*

b.

((u – 3\*dx) / (3\*dx\*(x + dx)) \* dx) + y = y\_new

Start constraints:

x1,1 = 1, x2,1 = 1, xn,1 = 0 for all n > 2

xi,2 = 0 for all n

x3,3 = 1, x4,3 = 1, xn,3 = 0 for all n != 2, 4

xn,4 = 0 for all n

x5,5 = 1, xn,5 = 0 for all n != 5

x6,6 = 1, xn,6 = 0 for all n != 6

xn,7 = 0 for all n

xn,8 = 0 for all n

x7,9 = 1, xn,9 = 0 for all n != 7

xn\_10 = 0 for all n

x8,11 = 1, xn\_11 = 0 for all n != 8

Objective function – Minimize the start time of the n-th operation

F(n) = 2\*xn,1 + 2\*xn,3 + xn,5 + xn,6 + xn,9 + xn,11

Resource constraints:

2 Units, 1 ALU, 1 mult/divide/compare

ALU:

x1,1 <= 1

x4,3 <= 1

x5,5 <= 1

x8,11 <= 1

Mult/divide/compare:

x2,1 <= 1

x3,3 <= 1

x6,6 <= 1

x7,9 <= 1

Problem 5:

a.

|  |  |  |  |
| --- | --- | --- | --- |
| x | Q | s | z |
| 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 |

Markov chain for toggle flip flop  
\*attached\*

S0 = probability of being in state 0  
S1 = probability of being in state 1

S0 = (1 – X)\*S0 + X\*S1  
S1 = (1 – X)\*S1 + X\*S0

S0 = S0 – X\*S0 + X\*S1

S0(1 – 1 + X) = X\*S1

S0/S1 = X/X

S0 = S1

S0 = 1 – S1 because you can either be in state 0 or state 1

Therefore…

S0 = ½

S1 = ½

Z(X) = X\*S1

Because for AND gate, the only time Z will be true is when we are in the 1 state and we have an X input coming in as a 1. Therefore X\*probability(S1) gives you the probability of Z

Z(X) = X\*(1/2) = X/2

b. Z(X,Y)

Markov chain for toggle flip flop

\*attached\*

S0 = probability of being in state 0

S1 = probability of being in state 1

S0 = S0(1 – X)(1 – Y) + S0(X)(Y) + S1(1 – X)(Y) + S1(1 – Y)(X)

S1 = S1(1 – X)(1 – Y) + S1(X)(Y) + S0(1 – X)(Y) + S0(1 – Y)(X)

S0 = S0(1 – X – Y + XY) + S0(XY) + S1(Y – YX) + S1(X – YX)

S1 = S1(1 – X – Y + XY) + S1(XY) + S0 (Y – YX) + S0(X – YX)

S0 = S0 – S0X – S0Y + S0XY + S0XY + S1Y – S1XY + S1X – S1XY

S0 – S0 + S0X + S0Y – S0XY – S0XY = S1Y – S1XY + S1X – S1XY

S0 (1 – 1 + X + Y – XY – XY) = S1(Y – XY + X – XY)

S0 (X + Y – 2XY) = S1 (X + Y – 2XY)

S0/S1 = (X + Y – 2XY)/(X + Y – 2XY)

Same probabilities as before, which makes sense

S0 = S1

S0 = 1 – S1

S0 = 1/2

S1 = 1/2

Equation for mux: i1\*s + i0\*s’

Z(X) = (X XOR Y)(X XOR Y)(S1) + (X XNOR Y)(Y)

X XOR Y = X’Y + XY’ = (1 – X)(Y) + (X)(1 – Y) = Y – XY + X – XY = X + Y – 2XY

X XNOR Y = XY + X’Y’ = (X)(Y) + (1 – X)(1 – Y) = XY + 1 – X – Y – XY = 1 – X – Y

(X XOR Y)(X XOR Y) = (X + Y – 2XY)(X + Y – 2XY) = 4X^2Y^2 – 4X^2Y + X^2 – 4XY^2 + 2XY + Y^2

(4X^2Y^2 – 4X^2Y + X^2 – 4XY^2 + 2XY + Y^2)\*(1/2) = 2X^2Y^2 – 2X^2Y + X^2/2 – 2XY^2 + XY + Y^2/2

(X XNOR Y)(Y) = (1 – X – Y)(Y) = Y – XY – Y^2

2X^2Y^2 – 2X^2Y + X^2/2 – 2XY^2 - Y^2/2 + Y