Computer architecture cleous mccalla biology essay

Science, Biology

February 4, 2014Assessment # 4Derive write both sum of products (SOP) and product of sums (POS) Boolean expressions describing the Output: Sum of products (SOP):

A

B

C

OUTPUT

EXPRESSION

0001

A’ B’ C’

00100101

A’ B C’

011010001011

A B’ C

1101

A B C’

1110SOP = (A’ B’ C’) + (A’ B C’) + (A B’ C) + (A B C’)Products of Sum (POS):

A

B

C

OUTPUT

EXPRESSION

00010010

A + B + C’

01010110

A + B’ + C’

1000

A’ + B + C

101111011110

A’ + B’ + C’

POS = (A + B + C’) • (A + B’ + C’) • (A’ + B + C) • (A’ + B’ + C’)Sum of products (SOP):

A

B

C

OUTPUT

EXPRESSION

0001

A’ B’ C’

0011

A’ B’ C

01000111

A’ B C

10001011

A B’ C

1101

A B C’

1111

A B C

SOP = (A’ B’ C’) + (A’ B’ C) + (A’ B C) + (A B’ C) + (A B C’) + (A B C)Products of Sum (POS):

A

B

C

OUTPUT

EXPRESSION

000100110100

A + B’ + C

01111000

A’ + B + C

101111011111POS = (A + B’ + C) • (A’ + B + C)Determine the sum of products expression for the following function: f(A, B, C) = (0, 1, 2, 6)If22= 421 = 220 = 1The binary values for the following are: 0= 0 0 01= 0 0 12= 0 1 06= 1 1 0Sum of products (SOP):

A

B

C

OUTPUT

EXPRESSION

0

0

0

1

A' B' C'

0

0

1

1

A' B' C

0

1

0

1

A' B C'

011010001010

1

1

0

1

A B C’

1110SOP = (A’ B’ C’) + (A’ B’ C) + (A’ B C’) + (A B C’)Assume that X consists of 3 bits, X2, X1, X0. Write a logic function that is true if and only if X contains only one 1.

X2 X1 X0.

Out

0 0 0

0

0 0 1

1

0 1 0

1

0 1 1

0

1 0 0

1

1 0 1

0

1 1 0

0

1 1 1

0F= X2’ X1’ X0+X2’ X1 X0’+X2 X1’ X0’Give the ASCII code for the letters U and k. ASCii Code for Upper case U= 85ASCii Code for lower case k= 107Why is a flip-flop also called a bistable? A flip-flop circuit is called bistable because they are digital logic circuits that can be in one of two stable states. They will maintain their state indefinitely until an input pulse called a trigger is received. How does a SR latch differ from a gated SR latch? The difference between a SR latch and a Gated SR Latch is that a SR latch will change its state whenever a change is made to the S or R input, but a Gated SR latch will only allow a change in state when the gate (denoted by E below) is high.

SR Latch

http://sub. allaboutcircuits. com/images/04173. png

Gated SR latch

http://sub. allaboutcircuits. com/images/04178. pngList any eight rules of Boolean algebra simplification. Eight rules of Boolean algebra simplification are: The Cumulative ruleExample: A+B = B+AThe Associative ruleExample: A + (B + C) = (A + B) + CThe Null ruleExample: A + 1 = 1 & A • 0 = 0The Absorption ruleExample: A + AB = AThe Idenpotency ruleExample: A + A = A & A • A = aThe Distributive ruleExample: A(B + C) = AB + ACThe

Adjacency ruleExample: AB + A’B = AUse either the rules of Boolean algebra or Karnaugh maps to simplify the following:

F = A + ABC +A’C

The karnaugh map table was populated based on the following factorsHighs (1s) for all the outputs where A is 1, no matter what B & C areHighs (1s) for all the outputs where A, B & C are all 1sHighs (1s) for all the outputs where A is 0 and C is 1 no matter what B isLows (0s) for all other outputs

A B

C

0

1

0 0

01

0 1

01

1 1

11

1 0

11Once A is High (1) the output is high no matter what B & C areOnce C is High (1) the output is high no matter what A & B areHence the simplified form of this expression is: F = A + CF = (A’B + C)’ + CFor this simplification we use Boolean algebra(A’B + C)’ + C=[(A’B)’ • C’] + C= [(A’’ + B’) • C’] + C=[(A + B’) • C’] + C= AC’ + B’C’ + C

NOT the entire expression:

(AC’ + B’C’ + C)’=(AC’)’ • (B’C’)’ • C’=(A’+C’’) • (B’’+C’’) • C’=(A’+C) • (B+C) • C’=(A’+C) • (BC’+ C’C)=(A’+C) • (BC’)=(A’B C’) + (BC’C)=(A’B C’) + (0)=(A’B C’)

NOT the entire expression a second time to get back to the original state:

(A’B C’)’= A’’+ B’ + C’’= A + B’ + CHence the simplified form of this expression is: F = A + B’ + C

F = (A + B)(A’ + B’)

For the first step of this simplification we use Boolean algebra(A + B) • (A’ + B’)= AA’ + AB’ + A’B + BB’ = AB’ + A’BWe construct a truth table to further simplify this expression

A B

Out

0 0

0

0 1

1

1 0

1

1 1

0Based on the operation of the above truth table we conclude that this is an Exclusive OR gate and hence the simplified expression is F= A oplus BDraw the logic gate equivalent to the following formulas: AB+C(A’ +B)’

A+B+C(A’+C’)

(A+B’)(BC +A)+D