Week 7: Assignment 7

The due date for submitting this assignment has passed.

Due on 2021-09-15, 23:59 IST.

As per our records you have not submitted this assignment.

1) Consider the code-snippet given below: ***2 points***

Let us assume that the array A is mapped to a Block RAM. Given one addition takes a single cycle and two adders are available, which of the following statements are correct? Assume that no source-code modification is done. Also, ignore the time needed to modify the loop iterator i. Only consider the time needed to execute the loop body.

S1: A single-port RAM will take 10 cycles to execute the entire module.

S2: A dual-port RAM will take 8 cycles to execute the entire module.

```
void fn(int A[20])
{
    for(i = 18; i < 20; i++)
        Sum[i] = A[i] + A[i-1] + A[i-2] + A[i-3];
}
```

- Both S1 and S2 are correct
- Only S1 is correct
- Only S2 is correct
- Neither S1 nor S2 is correct

No, the answer is incorrect.
Score: 0

Accepted Answers:
Both S1 and S2 are correct

2) While merging multiple arrays vertically, extra locations on the RAM are : ***1 point***

- padded with 1's.
- padded with 0's.
- padded with random values.
- not padded.

No, the answer is incorrect.
Score: 0

Accepted Answers:
padded with 0's.

3) Consider the code-snippets given below: ***1 point***

Assume that A and B are single-port RAMs and C is a dual-port RAM, all having width 32 bits where 32 bits is the size of an integer. Code snippet 2 is obtained by combining A and B into C. What is the optimization done here?

Code snippet 1:

```
int A[1000], B[200];
for(i = 1 to 1000)
{
    A[i] = value1;
    if(i<200)
        B[i] = value2;
}
```

Code snippet 2:

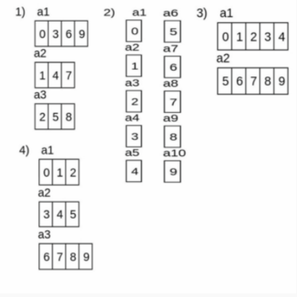
```
int C[1200];
for(i = 1 to 1000)
{
    C[i] = value1;
    if(i<200)
        C[i+1000] = value2;
}
```

- Array partitioning
- Array Horizontal merging
- Array vertical merging
- Array access permutation

No, the answer is incorrect.
Score: 0

Accepted Answers:
Array Horizontal merging

4) Consider an array A is partitioned into multiple small arrays. Which of the following represents correct Block Array Partitioning? ***1 point***



- 1) only
- 1) and 4)
- 3) only
- 1), 3) and 4)

No, the answer is incorrect.
Score: 0

Accepted Answers:
3) only

5) Initiation interval in loop-pipelining is: ***1 point***

- The interval in which outputs are produced
- Number of cycles needed to process one set of inputs.
- Number of cycles between the start of two consecutive iterations.
- Number of cycles between two outputs.

No, the answer is incorrect.
Score: 0

Accepted Answers:
Number of cycles between the start of two consecutive iterations.

6) Consider the code below. Assume that only one Single Port Block RAM R1 of size 1K and the word size of 70 bits is available. How can you map the arrays A1 and A2 to R1 so that the below code can be executed using the RAM R1? Select the appropriate option. ***2 points***

```
int A1[1000], A2[500];
for ( i = 0; I < 1000; i++)
{
  A1[i] = some data;
  if (i<500)
    A2[i] = some data;
}
```

- Merge A1 and A2 vertically to RAM R1
- Merge A1 and A2 horizontally to RAM R1
- Partition RAM R1 to two RAMs vertically
- Partition RAM R1 to two RAMs horizontally

No, the answer is incorrect.
Score: 0

Accepted Answers:
Merge A1 and A2 vertically to RAM R1

7) Consider a loop in which each iteration of the loop is scheduled in 4 time steps. The loop is implemented in fully pipelined mode with initiation interval 1. The loop has 20 iterations. How many time steps is required to execute the loop body? ***1 point***

- 20
- 80
- 21
- 23

No, the answer is incorrect.
Score: 0

Accepted Answers:
23

8) A loop in a C program is synthesized to RTL by HLS tool. Which of the following optimization are applied on Loop by the HLS tool? Select the best possible option. ***1 point***

- Loop Unrolling
- Loop pipelining
- Rolled implementation of Loop
- All of the above

No, the answer is incorrect.
Score: 0

Accepted Answers:
All of the above

Week 8: Assignment 8

The due date for submitting this assignment has passed.

Due on 2021-09-22, 23:59 IST.

As per our records you have not submitted this assignment.

1) Which of the below code segments will be synthesizable by Vivado HLS? (Remember: main() function is used for testing the function to be synthesized and usually not synthesized to hardware) ***2 points***

I.

```
#include <stdio.h>
void test(int d[10])
{
      int acc = 0;
      int i;
      for (i=0;i<10;i++)
      {
            acc += d[i];
            d[i] = acc;
      }
}
#ifndef __SYNTHESIS__
int main ()
{
      int d[10], i;
      for(i=0;i<10;i++)
      {
            d[i] = i;
      }
      test(d);
      for(i=0;i<10;i++)
      {
            printf("%d %d\n", i, d[i]);
      }
      return 0;
}
#endif
```

II.

```
#include <stdio.h>
void test(int d[10])
{
      int acc = 0;
      int I, x;
      for (i=0;i<10;i++)
      {
            acc += d[i] + fscanf("&d", x);
            d[i] = acc;
      }
}

int main ()
{
      int d[10], i;
      for(i=0;i<10;i++)
      {
            d[i] = i;
      }
      test(d);
      for(i=0;i<10;i++)
      {
            printf("%d %d\n", i, d[i]);
      }
      return 0;
}
```

III.

```
unsigned foo (unsigned n)
{
      if (n == 0 || n == 1) return 1;
      return (foo(n-2) + foo(n-1));
}
```

IV.

```
#include <stdio.h>

int main ()
{
      int d[10], i;
      for(i=0;i<10;i++)
      {
            d[i] = i;
      }
      test(d);
      for(i=0;i<10;i++)
      {
            printf("%d %d\n", i, d[i]);
      }
      return 0;
}
```

- I
- II
- III
- IV

No, the answer is incorrect.
Score: 0

Accepted Answers:
I

2) What is the least number of Array Accesses for this particular code after array access is minimized for the array A by code rewriting? Assume that array A is stored in single port BRAM at interface. ***0 points***

```
void top (int A[], int **n)
{
      for( i = 1; i < 5; i++)
      {
            n[i] = A [i - 1] + A [i + 1] + A [i + 2] ;
      }
}
```

- 6
- 7
- 8
- 9

No, the answer is incorrect.
Score: 0

Accepted Answers:
7

3) Which of these is anti-dependency? ***1 point***

- Read after Write (RAW)
- Write after Read (WAR)
- Read after Read RAR
- Write after Write

No, the answer is incorrect.
Score: 0

Accepted Answers:
Write after Read (WAR)

4) False dependencies can restrict: ***1 point***

- loop tiling
- loop pipelining
- loop unrolling
- array partitioning

No, the answer is incorrect.
Score: 0

Accepted Answers:
loop pipelining

5) Consider the two statements given below: ***1 point***

S1: Dynamic memory allocation is not supported in HLS tools because the dynamic needs of memory cannot be supported in a fixed architecture like FPGA or ASIC.

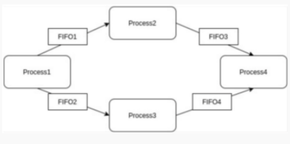
S2: System calls are not supported in HLS because there is no operating system for dedicated architectures.

- Both S1 and S2 are true.
- Only S1 is true.
- Only S2 is true.
- Neither S1 or S2 is true.

No, the answer is incorrect.
Score: 0

Accepted Answers:
Both S1 and S2 are true.

6) Consider the figure below. The amount of memory required for these FIFO buffers is 1024 bytes. If they are replaced by PING PONG buffers, what will be the memory required? ***1 point***



- 1024 bytes
- 2048 bytes
- 512 bytes
- Can not be determined

No, the answer is incorrect.
Score: 0

Accepted Answers:
2048 bytes

7) Dataflow optimizations cannot be applied for: ***1 point***

- Single-producer-consumer violations.
- Bypassing tasks.
- Feedback between tasks.
- All of the above scenerios.

No, the answer is incorrect.
Score: 0

Accepted Answers:
All of the above scenerios.

8) Which of the below type qualifiers cannot be synthesized by High-Level Synthesis? ***1 point***

- static
- const
- pointer
- None of the above

No, the answer is incorrect.
Score: 0

Accepted Answers:
None of the above

Week 9: Assignment 9

The due date for submitting this assignment has passed.

Due on 2021-09-29, 23:59 IST.

As per our records you have not submitted this assignment.

1) Which transformation has been applied in the below code snippet *Optimized code*? *1 point*

Source code:

```
do i = 1, n
    a[i] = a[i] + sqrt(x);
end do
```

Optimized Code:

```
if(n > 0) C = sqrt(x);
do i = 1, n
    a[i] = a[i] + C;
end do
```

- Constant Folding
- Code Motion
- Tree Height Reduction
- Copy Propagation

No, the answer is incorrect.
Score: 0
Accepted Answers:
Code Motion

2) Consider the code-snippets below and the statements that follow: *1 point*

S1: C2 is more efficient than C1 because it increases locality of reference of array accesses.

S2: C2 may not be more efficient than C1 because it changes the iteration order

C1.
```
for (i = 0; i < N; i++)
    for (j = 0; j < N; j++)
        y[i] += A[i][j] * x[j];
```

C2.
```
for (ii = 0; ii < N; ii += B)
    for (jj = 0; jj < N; jj += B)
        for (i = ii; i < ii+B; i++)
            for (j = jj; j < jj+B; j++)
                y[i] += A[i][j] * x[j];
```

- Both S1 and S2 are true.
- Only S1 is true.
- Only S2 is true.
- Neither S1 or S2 is true.

No, the answer is incorrect.
Score: 0
Accepted Answers:
Only S1 is true.

3) Consider the code snippets given below and select the correct option related to which transformations are applied on the Source Code to obtain the Transformed Code. *0 points*

Source Code:

```
for(i = 0; i < 2; i++)
    for(j = 0; j <2; j++)
        a[i][j] = a[i][j] / 8;
```

Transformed Code:

```
for(i = 0; i < 2; j++)
    for(i = 0; j <2; i++)
        a[i][j] = a[i][j] >> 3;
```

- Only Loop Interchange.
- Only operator strength reduction.
- Both Loop Interchange and Operator Strength Reduction
- None of the above

No, the answer is incorrect.
Score: 0
Accepted Answers:
Both Loop Interchange and Operator Strength Reduction

4) Will the below code transformation result in a performance improvement in terms of latency? *1 point*

Source Code:

```
for(i = 0; i < 99; i++)
    for(j = 0; j < 99; j++)
        x[i][j] = x[i - 1][j+1];
```

Transformed Code:

```
for(j = 0; j < 99; j++)
    for(i = 0; i < 99; i++)
        x[i][j] = x[i - 1][j+1];
```

- No, since the transformation is incorrect.
- No, since the transformation has no impact
- Yes, since the transformation results in performance improvement.
- None of the above.

No, the answer is incorrect.
Score: 0
Accepted Answers:
No, since the transformation is incorrect.

5) Following code-snippet is an example of: *1 point*

- Variable Propagation
- Copy Propagation
- Variable Renaming
- Code Motion

No, the answer is incorrect.
Score: 0
Accepted Answers:
Variable Renaming

6) Consider the matrix multiplication code snippet below. Assume N = 10. The maximum area overhead occurs when loop unrolling is applied on the : *1 point*

```
void multiply(int mat1[N][N], int mat2[N][N],
              int res[N][N]) {
 int i, j, k;
 for (i = 0; i < N; i++) {
  for (j = 0; j < N; j++) {
   res[i][j] = 0;
   for (k = 0; k < N; k++) {
    res[i][j] += mat1[i][k] * mat2[k][j];
}}} }
```

- Loop i
- Loop j
- Loop k
- None of the above

No, the answer is incorrect.
Score: 0
Accepted Answers:
Loop i

7) Consider the matrix multiplication code snippet again with N=10. If we apply Loop Pipelining Optimization on the inner most loop, which of the following statements is true compared to the iterative execution of the loop: *2 points*

S1: Design will run in faster clock if loop pipeline is applied as compared to its iterative execution.

S2: Area overhead is significant for loop pipeline implementation as compared to its iterative implementation.

```
void multiply(int mat1[N][N], int mat2[N][N],
              int res[N][N]) {
 int i, j, k;
 for (i = 0; i < N; i++) {
  for (j = 0; j < N; j++) {
   res[i][j] = 0;
   for (k = 0; k < N; k++) {
    res[i][j] += mat1[i][k] * mat2[k][j];
}}} }
```

- Only S1 is TRUE.
- Only S2 is TRUE.
- Both S1 and S2 are TRUE.
- Both S1 and S2 are FALSE.

No, the answer is incorrect.
Score: 0
Accepted Answers:
Only S1 is TRUE.

8) Consider the following function. Assume that the inputs a, b, c and d are 12 bit width integers. What would be the output data width? *1 point*

```
Return_type fn(int12 a, int12 b, int12 c, int12 d)
{
  intx t1, t2, t3;
  t1 = a + b;
  t2 =  c * d;
      t3 =  t1 * t2;

  return t3;
}
```

- 16
- 24
- 37
- 48

No, the answer is incorrect.
Score: 0
Accepted Answers:
37

9) Consider the following Merge Sort Code. Is it synthesizable by HLS tool? *1 point*

```
void mergeSort(int arr[], int l, int r)
{
  if (l < r) {
    // Same as (l+r)/2, but avoids overflow for
    // large l and h
    int m = l + (r - l) / 2;

    // Sort first and second halves
    mergeSort(arr, l, m);
    mergeSort(arr, m + 1, r);

    merge(arr, l, m, r);
  }
}
```

- Yes
- No

No, the answer is incorrect.
Score: 0
Accepted Answers:
No

Week 10: Assignment 10

The due date for submitting this assignment has passed.

Due on 2021-10-06, 23:59 IST.

As per our records you have not submitted this assignment.

1) Assume R_c represents the condition of execution and r_c represents the data transformation for a computation c in a finite state machines with datapaths (FSMD). Then, two computations c1 and c2 are equivalent if *1 point*

- $R_{c1} = R_{c2}$ and $r_{c1} != r_{c2}$
- $R_{c1} != R_{c2}$ and $r_{c1} = r_{c2}$
- $R_{c1} = R_{c2}$ and $r_{c1} = r_{c2}$
- $R_{c1} != R_{c2}$ and $r_{c1} != r_{c2}$

No, the answer is incorrect.
Score: 0
Accepted Answers:
$R_{c1} = R_{c2}$ and $r_{c1} = r_{c2}$

2) What would be the total number of computations for the FSMD generated from the code snippet given below? *1 point*

```
if(c1){
    if(c2)
        t1 = a + b;
    else
        t1 = c x d;
}
else{
    t1 = c x d;
}
```

- 1
- 2
- 3
- 4

No, the answer is incorrect.
Score: 0
Accepted Answers:
3

3) What would be the total number of paths and computations for the FSMD M1, respectively? Consider q0 as the reset state or start state and nodes with more than one outgoing edge as cut-points. (Note: A computation is an execution of the behaviour and a path is between two cut-points without having any intermediate cut-point) *2 points*

- 3, 2
- 6, 2
- 3, 8
- 6, 8

No, the answer is incorrect.
Score: 0
Accepted Answers:
6, 8

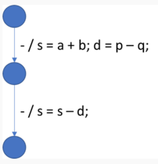
4) What would be the correct conversion of c code from the Verilog code given below. Consider a, c, x, y and m are registers. *2 points*

```
always @ (posedge ap_clk) begin
    if (1'b1 == ap_CS_fsm_state2) begin
    a <= b;
    c <= d;
    end
end
assign b = x + y;
assign d = a + m;
```

- ap_CS_fsm_state2;
  x_old = x;
  y_old = y;
  if (1 == ap_CS_fsm_state2)
  {
  a = x + y;
  }
  if (1 == ap_CS_fsm_state2)
  {
  c = a + m;
  }
- ap_CS_fsm_state2;
  b_old = b;
  d_old = d;
  if (1 == ap_CS_fsm_state2)
  {
  a = b_old;
  c = d_old;
  }
  if (1 == ap_CS_fsm_state2)
  {
  a = x + y;
  c = a + m;
  }
- ap_CS_fsm_state2;
  a_old = a;
  x_old = x;
  y_old = y;
  m_old = m;
  c_old = c;
  if (1 == ap_CS_fsm_state2)
  {
  a = x_old + y_old;
  }
  if (1 == ap_CS_fsm_state2)
  {
  c = a_old + m_old;
  }
- None of the above

No, the answer is incorrect.
Score: 0
Accepted Answers:
ap_CS_fsm_state2;
a_old = a;
x_old = x;
y_old = y;
m_old = m;
c_old = c;
if (1 == ap_CS_fsm_state2)
{
a = x_old + y_old;
}
if (1 == ap_CS_fsm_state2)
{
c = a_old + m_old;
}

5) Consider the following path. What would be the final symbolic value of s at the end of the path? *1 point*



- a + b
- s - d
- a + b - d
- a + b - p + q

No, the answer is incorrect.
Score: 0
Accepted Answers:
a + b - p + q

6) Consider the following two statements. Which one is TRUE about verification of High-level Synthesis? *1 point*

S1: Formal verification checks the equivalence of inputs and outputs of HLS symbolically; it does not need any test case.

S2: Simulation based verification of guarantees 100% correctness of the High-level Synthesis.

- Only S1 is True
- Only S2 is True
- Both S1 and S2 are True
- Both S1 and S2 are False

No, the answer is incorrect.
Score: 0
Accepted Answers:
Only S1 is True

7) Reverse engineering an equivalent C code from the HLS generated RTL helps in faster simulation based verification of HLS. *1 point*

- True
- False

No, the answer is incorrect.
Score: 0
Accepted Answers:
True

8) Which one of the following statements is correct? *1 point*

S1. RTL simulation is faster than C simulation.

S2. Phase wise verification of HLS needs intermediate information from the HLS tool.

S3. It is not possible to extract a C like behaviour from the RTL generated by the HLS tool.

- Only S1
- Only S2
- Only S3
- None of them

No, the answer is incorrect.
Score: 0
Accepted Answers:
Only S2

Week 11: Assignment 11

The due date for submitting this assignment has passed.

Due on 2021-10-13, 23:59 IST.

As per our records you have not submitted this assignment.

1) Consider the following statements: *1 point*

```
z = k1 ? ( x – y ) : ( y – x );
a = !k2 ? ( b * c ) : ( b / c );
t = k3 ? ( s – r ) : ( s + r );
```

Operations in the above statements are locked using a 3-bit key <k1 k2 k3>, which is < 1 0 0 > . What will be the operations performed for the given keys?

- z = y – x; a = b / c; t = s – r;
- z = x – y; a = b * c; t = s – r;
- z = x – y; a = b * c; t = s + r;
- z = y – x; a = b / c; t = s + r;

No, the answer is incorrect.
Score: 0
Accepted Answers:
*z = x – y; a = b * c; t = s + r;*

2) Consider the following statements: *1 point*

```
a = k1 ? ( b * c ) : ( b / c );
t = !k2 ? ( s + r ) : ( s – r );
x = !k3 ? ( y + z ) : ( y – z );
```

Operations in the above statements are locked using a 3-bit key < k1 k2 k3 > . For what values of keys the following operations mentioned below will be performed?

```
a = b / c;
t = s + r;
x = y – z;
```

- < 0 0 0 >
- < 0 0 1 >
- < 0 1 0 >
- < 1 1 0 >

No, the answer is incorrect.
Score: 0
Accepted Answers:
< 0 0 1 >

3) Consider the following piece of code: *1 point*

```
x = 20, y = 5;
if ( ( ( x – y ) > 10 ) XOR key1 )
{
    z = !key2 ? ( x / y ) : ( x – y );
}
else
{
    z = key2 ? ( x + y ) : ( x * y );
}
```

To get a final value of z as 100, what should be the key values < key1 key2 > ?

- < 0 0 >
- < 1 1 >
- < 0 1 >
- < 1 0 >

No, the answer is incorrect.
Score: 0
Accepted Answers:
< 1 0 >

4) Consider the following piece of code: *1 point*

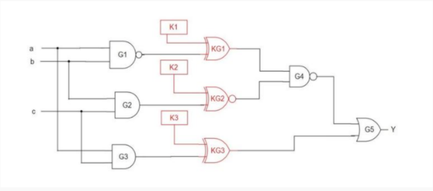
```
a = 10 , b = 2 , key1 = 1 , key2 = 0
if ( ( a > b) XOR key1 )
{
    c = key2 ? (a + b ) : ( a * b );
}
else
{
    c = !key2 ? ( a - b ) : ( a / b );
}
```

What will be the value of c at the end of the execution of the following code snippet?

- 12
- 20
- 8
- 5

No, the answer is incorrect.
Score: 0
Accepted Answers:
8

5) *1 point*



The circuit above is a locked circuit with three primary inputs a, b , c and three key inputs K1 , K2 , K3. G1 – G5 are primary gates and KG1 – KG3 are key gates. Which of the following can be a DIP, if key combinations 100 and 110 are considered?

- 001
- 111
- 110
- 101

No, the answer is incorrect.
Score: 0
Accepted Answers:
110

6) Consider the statement y = x + 5 . The constant '5' is to be locked and stored using a 4 bits register. What will be the locked constant value if the constant is locked using a 4-bit key < 1 1 0 1 > ? *1 point*

- 5
- 8
- 18
- 13

No, the answer is incorrect.
Score: 0
Accepted Answers:
8

7) Genuine IC obtained from the market is known as: *1 point*

- Good IC
- Oracle
- Correct Chip
- All of the above

No, the answer is incorrect.
Score: 0
Accepted Answers:
Oracle

8) Solution for IC overbuilding is known as: *1 point*

- Logic Blocking
- Circuit Locking
- Logic Locking
- All of the above

No, the answer is incorrect.
Score: 0
Accepted Answers:
Logic Locking

9) Consider the code snippet: *1 point*

```
if (a > b)
{
    z = x + y;
}
else
{
    z = x * y;
}
```

Suppose the condition has to be locked using a single bit key k whose value is 1. Choose the correct locked version of the code using the key k.

- if ((a > b) AND k) z = x * y; else z = x + y;
- if ((a > b) XOR k) z = x + y; else z = x * y;
- if ((a > b) XNOR k) z = x * y; else z = x + y;
- if ((a > b) XOR k) z = x * y; else z = x + y;

No, the answer is incorrect.
Score: 0
Accepted Answers:
*if ((a > b) XOR k) z = x * y; else z = x + y;*

10) A hardware circuit is locked using total k key bits. If the possible key combinations are 256, then what is the value of k ? *1 point*

- 9
- 8
- 7
- 6

No, the answer is incorrect.
Score: 0
Accepted Answers:
8

Week 12: Assignment 12

The due date for submitting this assignment has passed.

Due on 2021-10-20, 23:59 IST.

As per our records you have not submitted this assignment.

1) State whether the statement given below is true or false. LeFLow enables flexible FPGA High-Level Synthesis of Tensor Flow Deep Neural Networks. ***1 point***

- True
- False

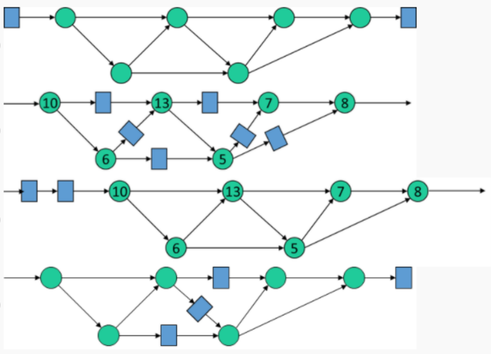
No, the answer is incorrect.
Score: 0
Accepted Answers:
True

2) State whether the statement given below is true or false. The biggest advantages of Halide are that it decouples the algorithm description of the program from the scheduling. ***1 point***

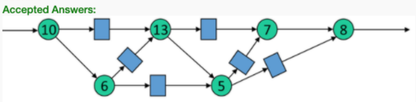
- True
- False

No, the answer is incorrect.
Score: 0
Accepted Answers:
True

3) Among four versions of a circuit, which one achieve fastest clock speed? The delay of the nodes are mentioned inside the nodes in options b and c. The delays are the same in all options. The circular nodes are combinational nodes and the rectangles are the registers. ***2 points***



No, the answer is incorrect.
Score: 0
Accepted Answers:

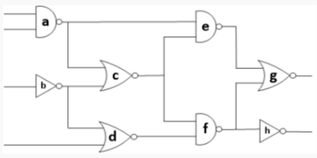


4) Which of the following statement is NOT true about folding transformation? ***1 point***

- Folding creates multiple synchronous clocks
- Folding increases register and MUX counts
- Folding of two functional units in series creates a feedback loop
- Folding of two functional units in parallel creates a feedback loop

No, the answer is incorrect.
Score: 0
Accepted Answers:
Folding of two functional units in parallel creates a feedback loop

5) Consider the following gate level design. Find the minimum 4 input LUTs required to implement it in FPGA? ***1 point***



- 1
- 2
- 3
- 4

No, the answer is incorrect.
Score: 0
Accepted Answers:
2

6) Let us assume that data path width is such that multiplier can be mapped to DSP block of FPGA. Which of the following statement is correct about DSP mapping? ***1 point***

S1: A multiplier with a pre-adder can be mapped to a single DSP.
S2: An accumulator for each multiplier cannot be mapped to a single DSP. The accumulator will be mapped to LUTs and the multiplier will be mapped to DSP block.

- S1
- S2
- S1 and S2
- None of the above

No, the answer is incorrect.
Score: 0
Accepted Answers:
S1

7) Which one of the following statements is NOT correct about Boolean functions? ***1 point***

- Boolean function can be implemented with only prime implicants.
- Essential prime implicant covers a minterm that is not covered by any other prime implicant.
- An irredundant cover is a cover that is not a proper superset of any cover.
- Don't care is an input combination that may or may not occur.

No, the answer is incorrect.
Score: 0
Accepted Answers:
Don't care is an input combination that may or may not occur.

8) Which of the following statements are true? ***1 point***

S1: Retiming is polynomial time solvable problem
S2: Timing of a design can be improved by retiming
S3: Area of a design can be optimized by retiming

- S1, S2 and S3
- S1 and S2
- S1 and S3
- S2 and S3

No, the answer is incorrect.
Score: 0
Accepted Answers:
S1, S2 and S3

9) Which one of the following methods is Not used for two level logic minimization? ***1 point***

- Karnaugh Map
- Quine–McCluskey method
- ESPRESSO
- Algebraic Model

No, the answer is incorrect.
Score: 0
Accepted Answers:
Algebraic Model