What is a hierarchy?

1

Define digital.

2

Define analogue.

3

How many values can be represented by a binary number containing n bits?

4

Arrange `AND`, `OR` *and* `NOT` *in order of operator precedence.*

5

What is the symbol for `AND`*?*

6

What is the symbol for `OR`*?*

7

What is the symbol for `NOT`

8

A hierarchy is a group of objects arranged in tiers of descending magnitude, importance or complexity.

1

An entity that can reside in one of two states at any one time.

2

An entity that can reside in an infinite number of possible states.

3

2^n

4

`NOT`, `AND`, `OR`

5

$\cdot$

E.g. $A \cdot B$

6

$+$

E.g. $A + B$

7

$\overline{}$

E.g. $\overline{A}$

8

What is the symbol for `XOR`

9

What is De Morgan's theorem commonly used for when designing digital circuits?

10

What is the symbol for an `AND` *gate?*

11

What is the symbol for an `OR` *gate?*

12

What is the symbol for an `XOR` *gate?*

13

What is the symbol for an `NOT` *gate?*

14

What is the symbol for an `NAND` *gate?*

15

What is the symbol for an `NOR` *gate?*

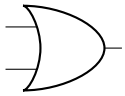
16

Converting gates such as `AND`, `OR`, `XOR` *etc into* `NAND` *and* `NOR` *since they are cheap and fast.*

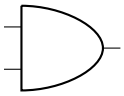
10

$\oplus$
E.g. $A \oplus B$

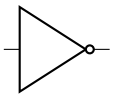
9



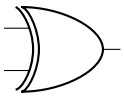
12



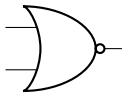
11



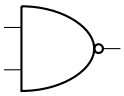
14



13



16



15

What's the symbol for a n:1 multiplexer?

17

What is the truth table for binary addition?

A	B	c_{in}	S	c_{out}
0	*0*	*0*		
0	*0*	*1*		
0	*1*	*0*		
0	*1*	*1*		
1	*0*	*0*		
1	*0*	*1*		
1	*1*	*0*		
1	*1*	*1*		

18

How do you negate a binary number?

19

Convert binary 6 *to* -6

20

Which bit is the signed bit when using 2's complement?

21

How do you subtract two binary numbers?

22

What is the sum-of-products?

23

What is the product-of-sums?

24

What is the truth table for binary addition?

A	B	c_{in}	S	c_{out}
0	*0*	*0*	*0*	*0*
0	*0*	*1*	*1*	*0*
0	*1*	*0*	*1*	*0*
0	*1*	*1*	*0*	*1*
1	*0*	*0*	*1*	*0*
1	*0*	*1*	*0*	*1*
1	*1*	*0*	*0*	*1*
1	*1*	*1*	*1*	*1*

18

17

1. Start with `0110`
2. Invert the bits - `1001`
3. Add 1 - `1010`

20

1. Invert the bits
2. Add 1

19

1. Invert the number you're subtracting
2. Add 1 to the inverted number
3. Add the number you're subtracting from with the inverted number.
Basically, add the original number to the 2's complement negative of what you're taking away.

22

The left most bit.

21

When a number of `OR` *gates are* `AND`*'ed together.*

24

When a number of `AND` *gates are* `OR`*'ed together.*

23

What is the structure of a half adder (in terms of gates)?

25

What is the truth table for the half adder?

A	B	S	c_{out}
0	*0*		
0	*1*		
1	*0*		
1	*1*		

26

Define propagation delay.

27

Fill in the table:

State	*Symbol*
Low	
High	
Tristate	
Unknown	

28

What does active high and active low mean?

29

What are the advantages of a hierarchy:

30

What is a combinatorial circuit?

31

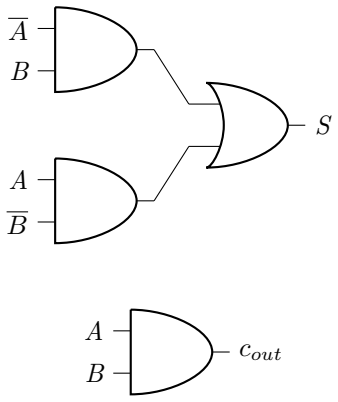
What is a sequential circuit?

32

What is the truth table for the half adder?

A	B	S	c_{out}
0	*0*	*0*	*0*
0	*1*	*1*	*0*
1	*0*	*1*	*0*
1	*1*	*0*	*1*

26



25

Fill in the table:

State	*Symbol*
Low	*0*
High	*1*
Tristate	*Z*
Unknown	*X*

28

Propagation delay is the time taken for the output of a gate to change after it's inputs have changed.

27

1. Encapsulation
2. Reuse of logic
3. Only have to define and test things once

30

If a signal is active high, then it is interpreted as `True` *when the signal is high (e.g. a light is on, or the voltage is positive etc).*
If a signal is active low, then it is interpreted as `True` *when the signal is low (e.g. a light is off, or the voltage is negative etc).*

29

A circuit where the value of the output depends on the values of the input and the past history of it's inputs. A sequential circuit requires a clock and memory.

32

A circuit where the value of the output depends only on the values of the input.

31

What is a synchronous clock?

33

What is a clock edge?

34

Define bistable.

35

What is a flip flop?

36

What is a register made of?

37

What is a finite state machine?

38

What do `S` *and* `R` *stand for on a S-R flip flop?*

39

What is the circuit symbol for a D-type latch?

40

The point on a clock signal where the signal is going from low to high (rising or positive edge) or high to low (falling or negative edge.)

34

A synchronous clock is one that is effective system wide; all components in the system adhere to this clock.

33

A bistable device that latches onto a state.

36

An entity that can be in one of two stable states.

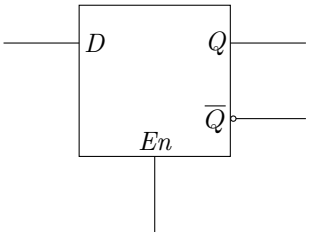
35

A digital system that holds the current state of itself and progresses to a new state based on the value of the current state.

38

A series of flip flops, each containing one bit.

37



40

`S` - *Set*
`R` - *Reset*

39

What is the structure of a D-type flip flop?

41

Is a D-type latch level sensitive or edge sensitive? What does that mean?

42

What is the circuit symbol for a D-type flip flop?

43

What does a D-type flip flop implement to make it synchronous?

44

What are the three delays in the D-type flip flop?

45

In order to ensure that a D-type flip flop will change state successfully, when does the input have to be constant?

46

What is the propagation delay?

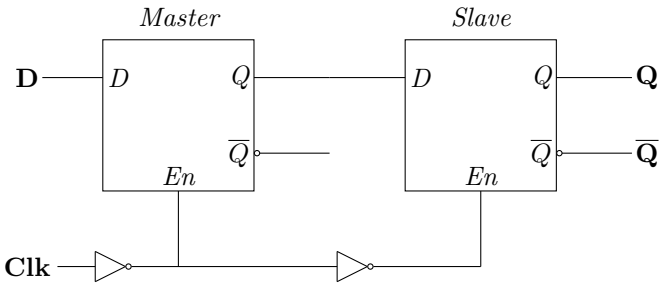
47

What is edge speed?

48

The D-type latch is level sensitive, meaning that it will change state whenever the input changes, this means it's **asynchronous**.

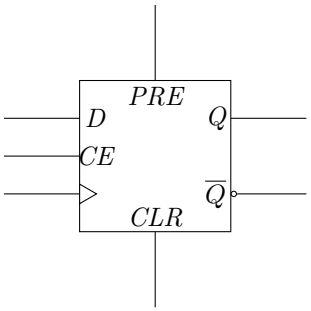
42



41

It has an enable switch that can be linked to the clock so that it will only change state when the device is clocked.

44



43

From the beginning of the set-up time to the end of the hold time.

46

The set-up time (T_{SU}), the hold time (T_H) and the propagation delay (T_{PD}).

45

The time taken for a signal to change state.

48

The time taken for the flip flop to output a change of state.

47

What does RTL mean?

49

Order the following in terms of their level of abstraction from lowest to highest:
· Logic Gate Level
· Transistor Level
· Register Transfer Level

50

What does the CE pin do on a register?

51

What are registers made up of?

52

Define the datapath.

53

What does the control block do at the Register Transfer Level?

54

What data flows between the datapath and the control block?

55

For a finite state machine with n states, how many flip flops are needed in the register?

56

1 *Transistor Level*
2 *Logic Gate Level*
3 *Register Transfer Level*

50

Register Transfer Layer

49

Lots of flip flops.

52

CE stands for clock enable, and when it is low, the register will ignore clock cycles.

51

The control block controls the operation of the datapath. It sends control signals to the datapath so that the data does indeed, take the right path.

54

The datapath is the path of registers and logical functions that the data flows through.

53

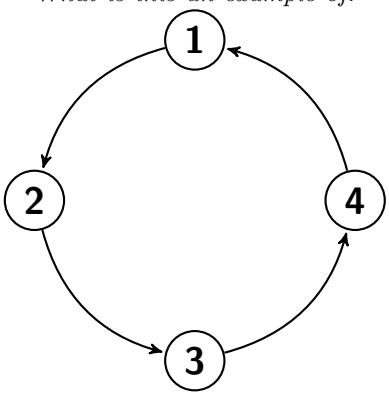
At least $\log_2 n$

56

The control block gives the datapath control inputs. The datapath gives control outputs to the control block.

55

What is this an example of:



57

What is the synchronous paradigm?

58

What is the formula to subtract A from B using 2's complement?

59

Name the advantages and disadvantages of a FPGA chip.

60

How do you cause a delay of $300ns$ *in a Verilog stimulus file? Can you use the same command in Verilog files describing hardware?*

61

How do you stop a simulation running in a Verilog stimulus file?

62

Give some advantages and disadvantages of using a Hardware Description Language (HDL).

63

How do you define a module in Verilog?

64

When all state changes in a system happen at once (usually at the same time as a clock transition).

58

A state transition diagram.

57

Advantages
· *Don't have to create an actual silicon chip to test hardware designs.*
· *Far faster than a software hosted simulation.*
Disadvantages
· *Slower operation than a custom chip.*
· *Requires more power than a custom chip.*
· *It has a lower capacity than a custom chip (fewer gates can be fit on to it).*

60

$$A - B = A + \overline{B} + 1$$

59

`$stop`

62

`#300`
You can't use that command in normal Verilog files since there is no way for the synthesiser to reliably create a delay of $300ns$ *in hardware.*

61

```
module <name> (input <input_1>, ..., <input_n>,
               output <output_1>, ..., <output_n>);
  //content
endmodule
```

64

Advantages
· *Is able to express some functions very easily in little code (e.g. addition or comparison)*
· *Will synthesise the gates for you, so you don't have to code the logic at the gate level.* · *The code will be easy to read if it's well structured.* **Disadvantages**
· *An inefficient circuit could be synthesised.*

63

Is Verilog case sensitive?

65

How are numbers formatted in Verilog?

66

How do you find the complement of a variable in Verilog?

67

What is required if you want to put more than one statement in a Verilog block?

68

What are the three forms of assignment in Verilog?

69

What will Verilog blocks containing continuous assignments synthesise into?

70

What is the syntax for an in-line conditional statement in Verilog?

71

What types of assignment can go in a always block?

72

`<bits>'<base><number>`
e.g. `4'b1010`, `8'd255`, `8'hFF`

66

Yes

65

```
begin
  statement_1
  ...
  statement_n
end
```

68

variable = ∼*variable*

67

Combinatorial logic.

Name	**Example**
Continuous	`assign varname = value`
Blocking	`varname = value`
Non-blocking	`varname <= value`

70

69

Either blocking or non-blocking assignments (but not both!).

`(cond) ? if_true : if_false`

72

71

What is the syntax for an always block?

73

Ensure all inputs on the [redacted] *of statements in an always block are in* [redacted].

74

What is the syntax for a Verilog case statement?

75

How do you define what edge of a signal an always block should be triggered on?

76

Non-blocking assignments happen [redacted].

77

In order to produce a FSM in Verilog, what do you write?

78

What does a diagram of a general FSM look like?

79

What differentiates between an always block that will be synthesised into a sequential circuit and one that will be synthesised into a combinatorial circuit?

80

Ensure all inputs on the Right hand side of statements in an always block are in the sensitivity list.

74

```
always @ (sensitivity_list)
begin
  ...
end
```

73

```
always @ (posedge clock)
```

or

```
always @ (negedge clock)
```

76

```
always @ (sel, w, x)
begin
  case(sel)
  0:  q = w;
  1:  q = x;
  default:  q = 0;
  endcase
end
```

75

```
always @ (posedge clock)
begin
 if (reset == 1) state <= 0;
 else
  case(state)
  0:  state = 1;
  1:  state = 2;
  2:  state = 0;
  default:  state = 0;
  endcase
end
```

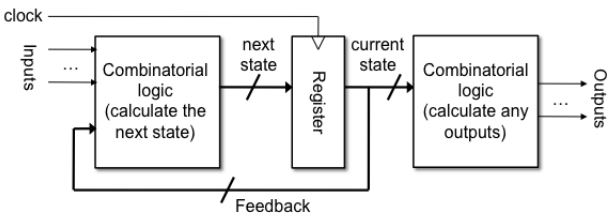
78

Non-blocking assignments happen simultaneously.

77

If the always block has a clock as an input, then it will be sequential.

80



79