Some more interview questions:
常见的题目有sequence detector, edge detector, moore/mealy machine, frequency divider, round robin arbiter, setup/hold等等
很少有面试官会问，问到也不会问很复杂的用法
熟练使用assertion来check简单的req，ack信号即可
func coverage知道怎么自定义bin，ignore_bin，cross coverage，transition coverage我觉得就够了

1. swap 2 variable without using temp (this is asked several times)

```
   int a = 5;
   int b = 10;

   a = a ^ b;  // a is a ^ b ,  b is b
   b = a ^ b;  // a is a ^ b ,  b is b ^ ( a ^ b ) = a ^ (b ^ b) = a ^ 0 = a
   a = a ^ b;  // a is (a ^ b) ^ a = ( a ^ a ) ^ b = 0 ^ b = b,  b is a already
```

  why use xor because using add will overflow (a = a + b; b = a - b; a = a - b;   will overflow for a+b)

2. reverse single linked list(iterative/recursively) leetcode problem

3. fibnacci series (iterative/recursive), some phone interview

4. determine if an integer is palindrome ("12321") , leetcode problem

5. reverse all bits of number in binary(0b111001 -> 0b100111), use bit manipulation, leetcode problem

6. find if a linked list has loop, leetcode problem

7. how do you write a fix priority arbiter? (I think most people can not find a good solution online, the following code comes from cornell course ECE5745)

```
    module fixed_arbiter #(parameter NUM_REQS=4)
    (
        input [NUM_REQS-1:0] req;
        output[NUM_REQS-1:0] grants;
    );
        wire[NUM_REQS:0] kills;
        assign kills[0] = 1'b0;

        wire[NUM_REQS-1] grants_int;
```

```
        genvar i ;
        generate
            for (i = 0; i < NUM_REQS; i++) begin: per_req_logic
                assign grants_int[i] = !kills[i] & reqs[i];
                assign kills[i+1] = kills[i] | grants_int[i];
            end
        endgenerate

        assign grants = grants_int;
    endmodule
```

8. how to write a round robin arbiter? (If you can understand item 7 how kill chain works, then it is easy to understand following code, you first need to have a variable priority arbiter, which takes an input of priority, then you can build the round robin arbiter with a shift register and the variable priority arbiter)

```
       module variable_priority_arbiter#(parameter NUM_REQS=4)
       (
           input [NUM_REQS-1:0] priority,   // one hot input of variable
priority
           input [NUM_REQS-1:0] reqs,
           output [NUM_REQS-1:0] grants
       );
           /*
               suppose the input priority is 00100
               priority_int will be          00000 00100

               imagine the reqs_int  is  01000 01000    //case 1
               imagine the reqs_int  is  00100 00100    //case 2
               imagine the reqs_int  is  00010 00010    //case 3
           */
           wire [2*NUM_REQS:0] kills;
           wire [2*NUM_REQS-1:0] priority_int = { {NUM_REQS{1'b0}}, priority
}; //extend priority
           wire [2*NUM_REQS-1:0] reqs_int = {reqs,reqs}; //copy reqs
           wire [2*NUM_REQS-1:0] grants_int;

           assign kills[0] = 1'b0;

           genvar i;
           generate
               for (i = 0; i < 2*NUM_REQS; i=i+1) begin: per_req_logic
                    assign grants_int[i] = priority_int[i] ? reqs_int[i]:
(!kills[i] & reqs_int[i]);
                    assign kills[i+1] = priority_int[i]? grants_int[i] :
(kills[i] | grants_int[i]);
               end
           endgenerate
```

```
          assign grants = (grants_int[NUM_REQS-1:0] | grants_int[2*NUM_REQS-
1:NUM_REQS];
      endmodule

      module round_robin_arbiter#(parameter NUM_REQS=4, paramater
RESET_PRIORITY = 1)
      (
          input clk,
          input rst,
          input [NUM_REQS-1:0] reqs,
          output [NUM_REQS-1:0] grants
      );
          wire priority_en = |grants;

          wire [NUM_REQS-1:0] priority_next;

          assign priority_next = {grants[NUM_REQS-2:0],grants[NUM_REQS-1]};

          wire [NUM_REQS-1:0] priority_;

          // a reset_req with reset value as RESET_PRIORITY
          reset_reg#(NUM_REQS,RESET_PRIORITY) priority_req
          (
              .clk(clk), .reset(reset), .en(priority_en), .d(priority_next),
.q(priority_)
          );

          //instantiate variable arbiter
          variable_priority_arbiter#(NUM_REQS)  variable_arbiter
          (
              .priority(priority_), .reqs(reqs), .grants(grants)
          );
      endmodule
```

3% (36) 👎
Digital Logic:
This is another important part. Some design questions will be asked.
- boolean algebra, de morgans theory
- K-map
- arithmetic logic (half adder/full adder/ how to use full adder count no of 1's in 7 bit?carry ripple adder/comparator)
- how to use mux implement gate(or/not....)
- how to use NAND/NOR implement all gate not/or/and...? (use 4 NAND implement XOR, use 4 NOR implement XNOR)
- how to use tri-state buffer and not gate to implement all gate?
- state matchine, state reduction
- sequence detector (overlay/non-overlay)
  you can use state machine/shift register.

state machine, what is differece between Mealy/Moore state machine.
101/110/1001/1011/1010/1101/10010/101X1/10XX1 try yourself solve all of them with both state matchin/shift register
- setup time/ hold time, where do they come from? how to solve them. what is metastablity
- give you an inifinite sequence, you every 1 bit every cycle, write the state matchine if the current number can be divided by 5?
what if MSB coming first? what if LSB coming first?
- how to do a divide by 2 clk divide? how about divide by 3? how to make it 50% duty cycle? how to do a divide by 5 with 50% duty cycle?
- synchronizer (2 FF), toggle synchronizer(just google it)
- synchronous fifo code
- asynchronus fifo(there is a paper design and synthesis technique of asynchronous fifo just understand it, it use grey code)
- how to write a fix-priority arbiter, how to write a round robind arbiter (use kill chain). How do you verify it?
- google "ASIC interview puzzle", some people like use questions from it.
- how to write a CAM?
- how to design HW linked list

3% (36) 👎
continue
array_reduction(sum(), product(), and(),or(),xor())
example:

```
byte b[] = {1,2,3,4}
int d = b.sum() with (int'(item>=2))  // find total count of numbers bigger or equal to 2, notice you must have an int' casting
int d = b.sum() with (item>=2? item : 0) // sum of numbers bigger or equal to 2
logic bit_arr[1024]
int y =  bit_arr.sum() with (int'(item)); // without the casting, total result will be 0/1 ONLY!
```

suppose you have an associate array of object queue (m_object[$] m_associate_array[*]), how to pass this type as function argument? use typedef

- class
remember super.new() will always be called implicitly
polymorphism (same as C++, you'd better know virtual table pointer to explain how run time find the implementation)
difference between static task/ task static
difference between task/function
local/protected attribute, what is the usage
pure virtual class/pure virtual function
when to use class scope resolution operator(static element/static method/typedef/enum....)
parameterized class with example in LRM, typedef will help handling

parameterized significantly

- procedure
  initial block (you have have multiple initial block)
  always block
  final block

  how to generate a clock with initial block?

  reg clk
  real clock_period = *****;

  initial begin
      clk = 0;
      forever begin
          #(clock_period/2) clk = ~clk;
      end
  end

  automatic variable / static variable difference?
  fork join/join_none/join_any

  for (int i = 0; i <=10; i++) begin
      fork
          automatic int j = i;   //note you must declare a automatic copy of j here, otherwise it will be same value (10) when you spawn all some_task()
          begin
              print something;
              some_task(j);
          end
      join_none
  end

  another trick with static/automatic variable, following code will print 11ns 20, 20ns 20. Why ? Because task is static, even for the argument. how to fix this? use task automatic print(i)
  suppose in a module,
  task print(i);
      #10ns;
      $display("%s ns %d",$time,i);
  endtask

  initial begin
      fork
          begin
              #1ns;
              print(10);
          end
          begin

```
                #10ns;
                print (20);
            end
        join
    end
```

interstatement assignment/intrastatement assignment difference.

```
#5 a = b;
a = #5 b;
a = @(posedge clk) b
a = repeat (5) @(posedge clk) b
```

how to use disable fork/wait fork

fine grained process_control (process::self() functions await/status/kill/suspend/resume)

foreach usage(foreach A[i,,k]) //note you can skill something here

what is the difference between passing an object handle by reference and by value? (you can modify the object content in both case, but if you pass by reference, you can even reassign the object handle, while in pass by value case, you are playing with the copy of object handle. just remember the normal case of integer argument...) so there is a problem in C++, how do you modify a pointer function argument? use void my_func(int* &m_pointer)

- clocking block, read the chapter carefully, i think most people do not understand it good
  synchronize signals to clock for sampling/driving
  input/output skew

- interface, read the chapter
  why do we need virtual interface?

- semaphores/mailbox/event, know the operations

- assertion (this is a long long long long chapter in LRM, personally I donot have too much experience in writing assertions, but try some examples)
  google system verilog assertion, there is some tutorial in duolos dot com which should be enough for interview

- constraint (this is REALLY important, almost every onsite will be several questions)
  how to write a onehot/onecold constraint? do not use function. one cold is similar to onehot just add a "~"

```
        rand bit [7:0] a;
```

```
            constraint one_hot_cons{
                a & (a-1) ==0;
                a !=0;
            }
          how to write constraint to unique array element, do not use unique
in LRM 2012
            rand bit[31:0] a[100];
            foreach (a[i]) {
                foreach (a[j]) {
                     if (i<j) {
                         a[i] != a[j];
                     }
                }
            }

           how to write a constraint for 8 queen? board[8][8]
           how to write a constraint to constraint the size of dynamic array
           how to write a constraint for increasing array
           rand/randc difference
           how to implement randc without using randc
           solve...before, why do we need this?
           dist := :/ difference
           inside

        - functional coverage (read the LRM and try some real example)

        - DPI ( export/import DPI functions, how to compile a shared library?
how to solve DPI scope issue? )


3% (36)    👎
2. UVM
    I think most companies want you to have exp in UVM. So read the UVM
cookbook should be enough.
    - TLM1  ( I do not use TLM2 and nobody ask me anything about TLM2)
       uvm_.*port/imp/export  ( I think UVM cookbook has a good explain for
imp/export)
       imp is directing "export" the implementation in that uvm_component
       export is "export" the implementation for sub uvm_components inside
the current uvm_component

    - read through your own uvm projects on
       define uvm_sequence_item (req/rsp)
       writing
uvm_driver/uvm_sequencer/uvm_monitor/uvm_scoreboard/uvm_agent/uvm_monitor
       writing uvm_sequence ( What is UVM reactive sequence, my understanding
is stimulus will be based on response from driver,
put_reponse()/get_response(), also note you need to set_id_info() to guide it
from driver into correct sequence)
```

- remember all uvm_phase, which phase are task, which are function?
- what phase is top-down and what is bottom up?
- what is difference between uvm_config_db and uvm_resource_db
- why do we need uvm_factory? (for easy type overriding without modifying base code)
- what is a virtual sequence/sequencer, why do we need
- how to pass a virtual interface through uvm_config_db
- sequence layering (translation sequence)

Bascially for UVM questions I do not think they will be out of the scope of UVM cookbook. If you have time you can also read through the UVM class reference, but that might be way too much

3% (36) 👎
Leetcode
I only do linked list/array/hash table/bit manuplation/two pointer/binary search/simple DP/simple DFS,BFS for preparing HW interview

numbers i do
1/3/5/7/9/15/16/17/18/19/20/21/22/23/26/27/33/35/39/40/46/48/49/53/55/62/64/69/70/78/80/82/83/88/89/90/136/137/141/142/148/155/160/162/167/169/189/190/191/203/206/215/225/229/231/232/234/237/242/260/268/278/287/301/326/342/374/384//693