

CSE 313s

Selected Topics in Computer Engineering

Sheet 4

خلينا قبل ما نحل، نتكلم عن arrays في SystemVerilog

(1) Static Arrays

- يستخدمه في حالة إني عايز احط عدد من elements معروف مسبقاً هو كام، في مكان ما contiguous جوة الـ memory
- بتحدد size بتاعه في compile time
- طلما عرفته وحددت size خلاص مينفعش تغيره
- Verbose declaration: `int array_1 [0:7]; indices array, array` // بتحدد بداية ونهاية الـ array
- Compact declaration: `int array_2 [8];` // هنا بتحدد size بس وهو by default زي اللي فوق

(2) Dynamic Arrays

- يستخدمه في حالة إني عايز احط عدد من elements متغير، شوية يزيد شوية يقل، أو يكون مبني على input معين مثلًّا هعرفه في runtime، في مكان ما contiguous جوة الـ memory
- بتحدد size بتاعه في runtime
- تقدر تغير size بتاعه أثناء simulation
- يتعرفه عن طريق الـ new construct بالـ memory مكان empty square brackets
- Array Declaration: `int dyn_array [];`
- Memory Allocation: `initial begin
 dyn_array = new[8];
 //...
end`
- Array Extending: `initial begin
 //...
 dyn_array = new[16](dyn_array);
 //...
end`

(3) Associative Arrays

- الـ Associative Array مختلفة عن النوعين السابقين، هي معتبرش data structure، ولكنها أقرب للـ array، ولكنها أقرب لـ data places contiguous في الـ memory
- الـ data بتتخزن في sparse memory locations
- الـ indexing فيها مختلف، لأنه مش شرط يكون integer زي اللي فاتوا، ولكن ممكن يكون اي حاجة، وبنسميه key
- يتعرفه عن طريق أنك بتكتب الـ key بتاعة الـ data type جوة الـ square brackets، وبعدين اثناء الـ run time بتـ assign value لكل key براحتك
- key-value pair تكون عبارة عن array elements
- Array Declaration: `int assoc_array [string];`
- Wild Card Declaration: `int assoc_array [*];` // مش محدد معين، كله ينفع
- Memory Allocation: `initial begin
 assoc_array["gpa"] = 4;
 //...
end`

(4) Queues

- أنا صراحةً بشوف الـ Queue شبه الـ stack، مختلفة عن الـ arrays إنك تقدر تشيل element من index معين، وتحط واحد في index تاني وهكذا، وشبه الـ stack في الـ push والـ pop
- زي ما قولتلك تخيلها كـ stack، أو مكان في الـ memory عمال يـ expands وـ shrinks على حسب الطلب، تحط تشيل تعليمي اللي أنت عايزه
- يتعرفه عن طريق أنك بتكتب الـ dollar sign جوة الـ square brackets أثناء الـ run time بتـ insert push assign وبـ pop وـ delete

- Array Declaration: `int q_array [$];`

- Memory Allocation: `initial begin
 //...
 q_array = {6, 7, 5};
 q_array.insert(1, 9);
 q_array.push_front(12);
 q_array.pop_back;
 //...
end`

- Which of the following are true with respect to System Verilog arrays?
 - Associative arrays can be used when size of an array is not known as it can be built as key/value pairs.
 - Dynamic arrays are useful for contiguous collection of variables whose number keeps varying.
 - Dynamic arrays can be re-sized after size is allocated.
 - All of the above

- What will be the output of the code below?

```

1 module test;
2   bit [31:0] xyz[*]; // associative array declaration with wild card
3
4 initial begin
5   xyz["a"] = 40; // key = "a", value = 40
6   $display(xyz.num());
7 end
8 endmodule

```

- a. a b. 1 c. 40 d. 41

• num(): associative array اللي في الـ entries return عدد اللي في الـ entries
 • size(): صفر لو فاضية زي اللي فوق بس return

- Logic variables can have multiple drivers.

- a. True
 b. False

احنا بتتكلم عن 3 main data types مع كل data type multiple drivers

• wire: assign statement عادي, ومش هتديك compile error وده بيكون عن طريق إن اعمل لنفس الـ signal كذا

```
assign wire_y = 4; // wire_y = 0100
assign wire_y = 5; // wire_y = 0101
```

والقيمة النهائية لـ wire تكون على حسب simulator VCS مثلاً، بيسيب الـ bits المشتركة زي ما هي، والمختلفة بيطتها بـ x

أنا مثلاً هيطلعك كل الـ bits xxxx في حالة الـ multiple drivers questa

• reg: always block جوة الـ reg ممكن يتعملها multiple assignments

```
always@(*)
  reg_x = a;

always@(*)
  reg_x = b;
```

• logic: assign statement في المـ logic هيقبل always block جوة الـ logic ولكن بيزعق في حالة المـ logic multiple assignments

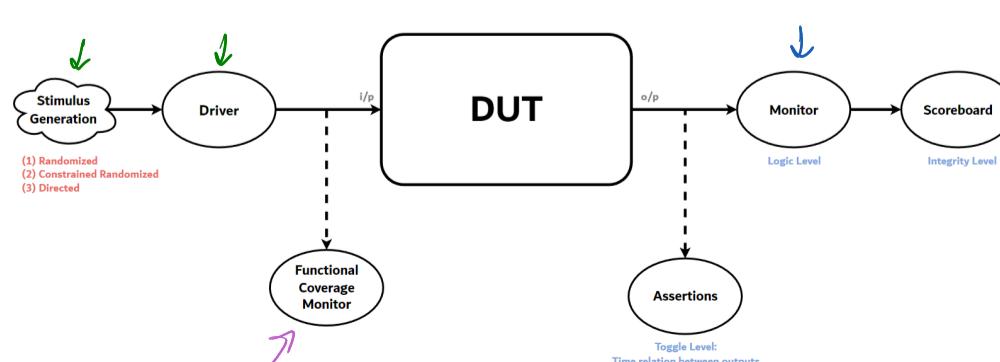
```
✓ always@(*)
  logic_z = a;
  X assign logic_z = a;
  assign logic_z = b;

always@(*)
  logic_z = b;
```

- 'bit' is two state
 - True
 - False

- Testbench functionality is to:

- Generate stimulus and apply to DUT
- Capture response and check for correctness
- Measure progress against verification goals
- All of the above

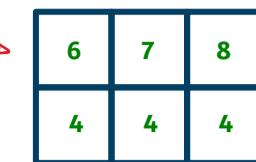


6. 'wire' variables can have multiple drivers

 - a. True
 - b. False

7. The declaration of the array shown in the following code is:

```
 {{6,7,8}, default:4};
```



- a. array_m[3][2]
 - b. array_m[1][2]
 - c. array_m[2][3] (Correct)
 - d. array_m[3][3]

8. What will be the output of the below code:

- a. Compilation error
 - b. `'{{5, 10, 7, 4, 3, 8}}`
 - c. `'{3, 4, 5, 5, 7, 7, 8, 10}`
 - d. `'{5, 10, 7, 4, 3, 5, 7, 8}`

```
module test;
  int array[8] = '{5,10,7,4,3,X,X,8};
  int que[$];
initial begin
  que = array.unique;
  $display(que);
end
endmodule
```

- **unique:** array فيه unique entries return queue بذاتي ما تقول كده بتاخذ array تحذف منه duplicated values وتحطه في queue وترجعهولك

9. With respect to dynamic array, which of the following are true (select all those apply)

- a. During runtime, array can be extended and retaining the old values queue and associative only

b. Can add and remove array elements from anywhere of the array ↗

c. Array can grow and shrink at runtime

d. Array is initially empty and space is allocated when new[] is called

10. Array [4][8]. This array declaration type is:

- a. Compact declaration
 - b. Verbose declaration
 - c. Single dimension array declaration
 - d. None of the above

11. Which of the following is queue declaration?

- a. int a[]; --> Dynamic Array
 - b. int a[\$];
 - c. int a; --> integer
 - d. int a[*]; --> Associative Array

12. What will be the output of the below code?

```
module test;
  int que[$];
  initial begin
    foreach(que[i]) begin
      $display("que element at index %0d = %0d", i, que[i]);
    end
  end
endmodule
```

- الـ code ده بيقول ايه؟

أولاً، حاجة عملت queueJ declaration

١٨١٦

س. ثانية واحدة!

- Fatal error
 - b. Que element at index 0 = 0 --> if it was initialized with ((que[0] = 0;))
 - c. Simulation ends without displaying display statement
 - Compilation error

13. Does SystemVerilog support multidimensional arrays?

- a. True
 - b. False

14. Is there any method to append one dynamic array to another?

yes, using concatenation

```
int array_1[];
int array_2[];
int array_3[];

initial begin
    array_1 = new[3];
    array_1 = '{1,2,3};

    array_2 = new[3];
    array_2 = '{4,5,6};

    array_3 = new[6];
    array_3 = {array_1,array_2};
end
```

15. Given a is an array defined as follows: `int a[] = {1,4,8,3,7,0,23,2,5,6};`

- **sum:** بتجمع كل الـ entries array وترجع نتيجة الجمـع
 - **find:** expressionJ satisfied elements فيه كل اللي return queue
 - **with clause:** casting 1-bit return expression بيـقي لازم تعمله sum في حالة الـ constraint على array method الذي يستخدمها، خـد بالـك إنه في النـهاية

What is the difference between the following statements?

(1) a.sum() with (item < 5)

هنا أنا حاطط expression بيشرط إن يجمع item يكون أقل من 5، ولكن فيه مشكلة! نتيجة الـ expression ده تكون `true` أو `false`, يعني bit-1 كده هيفضل يجمع bit-1 مع bit-1 مع.. ويحط الناتج في bit-1 بردو

يعني لو بصينا في الما، هنلاقي أن اللي بيتحقق condition ده هو {1,4,3,0,2} اللي هيحصل إنه هيحسب binary 101 يعني $1 + 1 + 1 + 1 + 1 = 5$ هو هي `return` وبأخذ LSB فقط وياخد truncate.

(2) `a.sum()` with `(int'(item < 5))`

ف في الحالة دي هي 5
ولكن السؤال، ايه الـ 5 دي؟

نتيجة الـ expression اللي حاطينه، هو كده بيجمع * عدد الـ elements * اللي بتحقق الشرط ده

طيب لو كنت عايزة جمع `elements` شخصياً اللي بتحقق الشرط مش عددهم؟ هنغير شوية في الـ `expression` ونخلية `item` return الـ `element` بدلاً من `true`

(3) a.find() with (item < 5)

16. What is the difference between Associative array and Dynamic array?

Associative Array

- Used for sparse memories and some scoreboards with a single index.
 - Uses key-value pairs for data storage
 - Keys can be of any data type, including strings, integers, or even user-defined types.

Dynamic Array

- A resizable array where the size can be changed dynamically during runtime.
 - Elements are stored in contiguous memory locations.
 - Accessed using indices starting from 0.

17. What's the difference between data type logic, req and wire?

reg

- It can be assigned ONLY inside Procedural Blocks

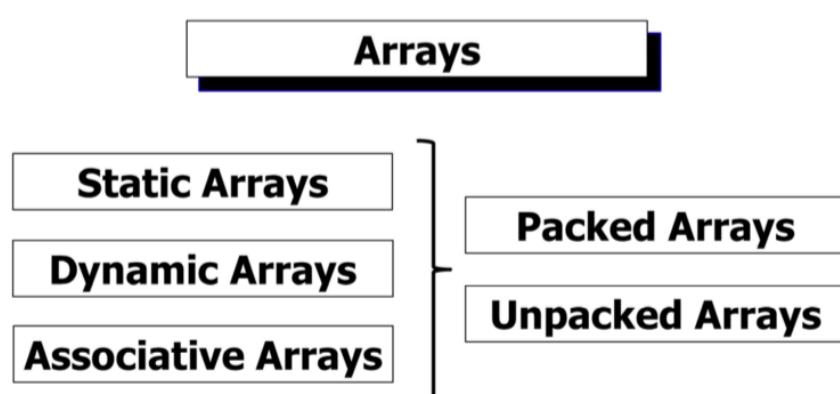
wire

- It can be assigned ONLY using Assign Statement

logic

- Introduced in SystemVerilog to address the ambiguity of reg and wire & can be assigned in both Procedural blocks and Assign Statements

18. How many array types in SystemVerilog? How do you use them? Show examples.



19. What is the difference between a bit and logic data type?

bit

- It's a two-state data type

logic

- It's a four-state data type
-

20. What is the difference between logic[7:0] and byte variable in SystemVerilog?

logic[7:0]

- It's a four-state data type, unsigned by default, No. of bits >= 1

byte

- It's a two-state signed data type, No. of bits = 8
-

21. Which of the array types: dynamic array or associative array, are good to model really large arrays, say: a huge memory array of 32KB?

Associative array are good to model really large arrays, as you will store entries in a sparse matrix and hold values in needed locations only.
But in case of dynamic array, you have to allocate the whole 32KB memory array while you will definitely access much fewer locations

22. Suppose a dynamic array of integers (myvalues) is initialized to values as shown below. Write a code to find all elements greater than 3 in the array using array locator method *find*?

```
int myvalues[] = {9, 1, 8, 2, 4, 6};
```

```
module sheet_4;  
  
int myvalues[] = {9, 1, 8, 3, 2, 4, 6};  
int que[$];  
  
initial begin  
    que = myvalues.find() with (item > 3);  
end  
endmodule
```

23. Given a dynamic array of size 20, how can the array be re-sized to hold 40 elements while the lower 20 elements are preserved as original?

```
module sheet_4;  
  
int dyn_array[];  
  
initial begin  
    dyn_array = new[20];  
  
    for(int i; i<20; i++)  
        dyn_array[i] = i;  
  
    dyn_array = new[40](dyn_array);  
end  
endmodule
```

Juba