

CSE 313s

Selected Topics in Computer Engineering

Sheet 4

خلينا قبل ما نحل, نتكلم عن arrays في الـ SystemVerilog:

(1) Static Arrays

• يستخدمه في حالة إني عايز احط عدد من الـ elements معروف مسبقًا هو كام, في مكان ما contiguous جوة الـ memory

• بتحدد الـ size بتاعه في الـ compile time

• طلما عرفته وحددت الـ size خلاص مينفعش تغييره

• **Verbose declaration:** `int array_1 [0:7];` // بتحدد بداية ونهاية الـ array, الـ indices

• **Compact declaration:** `int array_2 [8];` // هنا بتحدد الـ size بس وهو by default بيترجم زي اللي فوق

(2) Dynamic Arrays

• يستخدمه في حالة إني عايز احط عدد من الـ elements متغير, شوية يزيد شوية يقل, أو يكون مبني على input معين مثلاً هعرفه في الـ run time, في مكان ما contiguous جوة الـ memory

• بتحدد الـ size بتاعه في الـ run time

• تقدر تغير الـ size بتاعه اثناء الـ simulation

• بتعرفه عن طريق الـ empty square brackets وبعدين بتحجزله مكان في الـ memory بالـ new construct

• **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 متعبرش array, ولكنها أقرب للـ data structure, الـ data مش بتتخزن في الـ contiguous places في الـ memory

• الـ data بتتخزن في الـ sparse memory locations

• الـ indexing فيها مختلف, لأنه مش شرط يكون integer زي اللي فاتوا, ولكن ممكن يكون اي حاجة, وبنسميه key

• بتعرفه عن طريق أنك بتكتب الـ data type الـ key بتاعة الـ key جوة الـ square brackets أثناء الـ declaration, وبعدين اثناء الـ run time بت assign الـ value لكل key براحتك

• الـ array elements بتكون عبارة عن key-value pair

• **Array Declaration:** `int assoc_array [string];`

• **Wild Card Declaration:** `int assoc_array [*];` // مش محدد data type معين, كله ينفع

• **Memory Allocation:**

```
initial begin
    assoc_array["gpa"] = 4;
    //...
end
```

(4) Queues

• انا 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, او مكان في الـ memory عمال expand وي shrink على حسب الطلب, تحط تشيل عملي اللي انت عايزه

• بتعرفه عن طريق أنك بتكتب الـ dollar sign جوة الـ square brackets أثناء الـ declaration, وبعدين اثناء الـ run time بت assign الـ push وبت pop وبت insert وبت delete زي ما انت عايز

• **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
```

كانت مراجعة سريعة كده على أنواع arrays, تقدر ترجع للتفاصيل بقى في slides, خرينا نبدأ نحل

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

2. 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
```

- num(): associative array اللى في الentries اللى بتreturn عدد الentries اللى في الassociative array
- size(): زي اللى فوق بس بتreturn صفر لو فاضية

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

3. Logic variables can have multiple drivers.
- a. True
- b. False

احنا بنتكلم mainly عن 3 data types, خرينا نشوف كده ايه حكاية الmultiple drivers مع كل data type

• wire: بتسمح بالmultiple drivers عادي, ومش هتديك compile error وده بيكون عن طريق إن اعمل لنفس الsignal كذا assign statement

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

والقيمة النهائية للa هتكون على حسب الsimulator, بالنسبة لVCS مثلاً, بيسيب الbits المشتركة زي ما هي, والمختلفة بيحطها بx

أما questa مثلاً هيطلعلك كل الbits بxxxx في حالة الmultiple drivers

• reg: ممكن يتعملها multiple assignments جوة الalways block

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

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

• logic: الlogic هيقبل الmultiple assignments جوة الalways block ولكن بيزعق في حالة الmultipler drivers في الassign statement

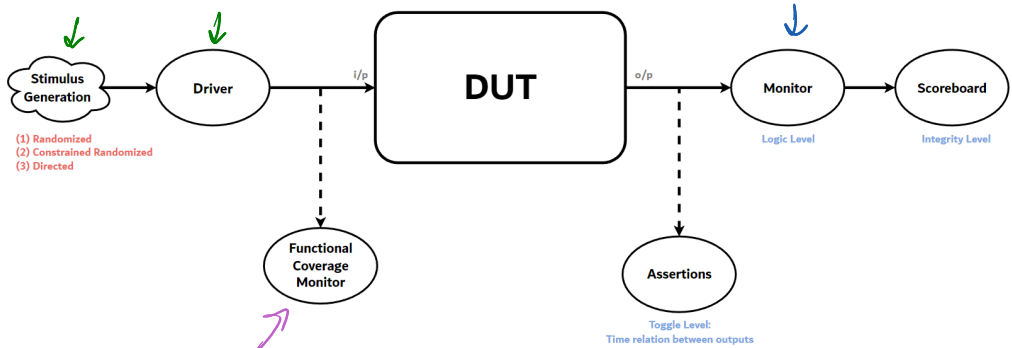
```
✓ always@(*)
  logic_z = a;

✗ assign logic_z = a;
  assign logic_z = b;

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

4. 'bit' is two state
- a. True
- b. False

5. Testbench functionality is to:
- ✓ a. Generate stimulus and apply to DUT
 - ✓ b. Capture response and check for correctness
 - ✓ c. Measure progress against verification goals
 - d. 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};
```

6	7	8
4	4	4

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

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

```
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: return queue فيه unique entries من array
زي ما تقول كده بتاخذ array تحذف منه duplicated values وتحطه في queue وترجعها

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
- 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
```

الcode ده بيقول ايه؟
أول حاجة عملت declaration لqueue
وبعدين دخلت في foreach loop على entries بتوع الqueue عشان أـdisplay كل واحد بالkey والvalue بتاعته
بس ثانية واحدة!
هو انا حطيت حاجة في الqueue أصلاً؟
أومال رايح تـdisplay ايه يا مولانا؟

- a. Fatal error
- b. Que element at index 0 = 0 --> if it was initialized with ((que[0] = 0;))
- c. Simulation ends without displaying display statement
- d. 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 array entries و return نتيجة الجمع
- **find:** بتعبر return فيه كل elements اللي satisfied ال expression
- **with clause:** ده زي constraint بحطه على array method اللي بستخدمها, خد بالك إنه في النهاية expression بي return 1-bit, يبقى لازم تعمله casting في حالة ال sum

What is the difference between the following statements?

(1) `a.sum() with (item < 5)` هنا أنا حاطط expression بيشتترط إن يجمع item يكون اقل من 5, ولكن فيه مشكلة! نتيجة ال expression ده هتكون true او false, يعني bit-1 كده هيفضل يجمع bit-1 مع bit-1 مع.. ويحط الناتج في bit-1 بردو

يعني لو بصينا في a, هنلاقي أن اللي بيحقق ال condition ده هو {1,4,3,0,2} اللي هيحصل إنه هيحسب $5 = 1 + 1 + 1 + 1 + 1$ يعني 101 ك binary هو هي truncate وياخد ال LSB فقط ويا return

(2) `a.sum() with (int'(item < 5))` هنا عالجت الأزمة اللي فاتت بال casting, ف كده هيجمع bits-32 مع bits-32 مع... ويحط الناتج في bits-32 بردو
ف في الحالة دي هي return 5
ولكن السؤال, ايه ال 5 دي؟
نتيجة ال expression اللي حاطينه, هو كده بيجمع *عدد ال elements* اللي بتحقق الشرط ده

طيب لو كنت عايز اجمع ال elements شخصيًا اللي بتحقق الشرط مش عددهم؟ هنغير شوية في ال expression ونخليه ي return ال element بدلًا من true `a.sum() with ((item < 5)? item:0)`

(3) `a.find() with (item < 5)` هت return queue فيه كل ال elements اللي بتحقق الشرط ده واللي قولنا عليهم مسبقًا: {1,4,3,0,2}

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

Associative Array	Dynamic Array
<ul style="list-style-type: none">• 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.	<ul style="list-style-type: none">• 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, reg and wire?

reg

- It can be assigned ONLY inside Procedural Blocks

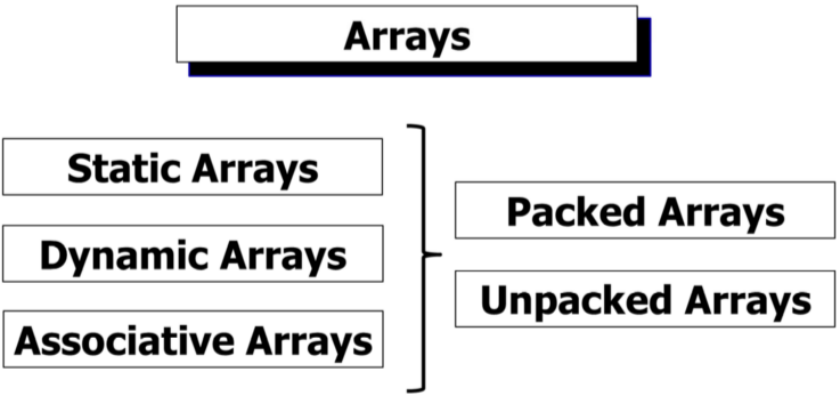
wire

- 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, 3, 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