

Queues

SystemVerilog introduces a new data type, the queue, which combines the best of an linked list and array. Like a linked list, you can add or remove elements anywhere in a queue, without the performance hit of a dynamic array that has to allocate a new array and copy the entire contents. Like an array, you can directly access any element with an index.

A queue is declared with word subscripts containing a dollar sign: [\$]. The elements of a queue are numbered from 0 to \$.

Syntax_Queue:

Data_type queue_name[\$];

-> queue capable of holding infinite elements.

CODE-1 if Queue has n elements and declare a queue that can store string value.

```
module queue;
string queue1[$];
initial begin
    queue1={"apple","cat","orange","Rat","cow"};
    foreach(queue1[i]) begin
        $display("\tqueue1[%0d]=%0d",i,queue1[i]);
    end
    $display("\nqueue1=%p",queue1); //in packed formate
end
endmodule
```

```
QuestaSim> do run.do
# QuestaSim-64 vlog 2021.1 Compiler 2021.01 Jan 19 2021
# Start time: 16:50:41 on Aug 31,2023
# vlog -reportprogress 300 queue1.sv
# -- Compiling module queue
#
# Top level modules:
#   queue
# End time: 16:50:41 on Aug 31,2023, Elapsed time: 0:00:00
# Errors: 0, Warnings: 0
# End time: 16:50:42 on Aug 31,2023, Elapsed time: 0:00:28
# Errors: 0, Warnings: 0
# vsim queue
# Start time: 16:50:42 on Aug 31,2023
# ** Note: (vsim-3813) Design is being optimized due to module recompilation...
# Loading sv_std.std
# Loading work.queue(fast)
#   queue1[0]=apple
#   queue1[1]=cat
#   queue1[2]=orange
#   queue1[3]=Rat
#   queue1[4]=cow
#
# queue1={"apple", "cat", "orange", "Rat", "cow"}
```

Types of queues in SystemVerilog

1. Bounded queue: Queue having a specific size or a limited number of entries.
2. Unbounded queue: Queue having non-specific queue size or unlimited entries.

SystemVerilog Queue methods

Size();

Number of elements in queue.

delete();

Delete element at the given index.

Insert();

Insert element at specific index of queue.

pop_front();

Pop the first element from the queue.

pop_back()

Returns and removes the last item of the queue.

push_front();

Inserts an item at the front of the queue.

push_back();

Inserts an item at the end of the queue.

shuffle();

shuffles items in the queue in random manner

CODE-2

declare queue of integer type and implemented queue method.

```
module queue_method;
integer intQ[$];

initial begin
    for(int i=0; i<=10; i++) begin
        intQ[i] = $urandom_range(50,60);
    end
    $display("\nqueue value=%p",intQ);
    $display("\nSize of the queue =%0d, \nintQ=%p",intQ.size(),intQ);
    intQ.delete(3);
    $display("\tAfter using intQ.delete(3)");
    $display("Size of the queue =%0d, \nintQ=%p",intQ.size(),intQ);

    intQ.pop_front();
    $display("After using intQ.pop_front()");
    $display("Size of the queue =%0d, \nintQ=%p",intQ.size(),intQ);

    intQ.pop_back();
    $display("After Using intQ.pop_back()");
    $display("Size of the oueue =%0d, \nintQ=%p",intQ.size(),intQ);

    intQ.push_front(10);
    $display("After using intQ.push_front(10)");
    $display("Size of the queue =%0d, \nintQ=%p",intQ.size(),intQ);

    intQ.push_back(17);
    $display("After using intQ.back_front(17)");
    $display("Size of the queue =%0d, \nintQ=%p",intQ.size(),intQ);

end
endmodule
```

```
-----
#
# queue value={57, 60, 58, 53, 55, 59, 53, 60, 56, 53, 59}
#
# Size of the queue =11,
# intQ={57, 60, 58, 53, 55, 59, 53, 60, 56, 53, 59}
#   After using intQ.delete(3)
# Size of the queue =10,
# intQ={57, 60, 58, 55, 59, 53, 60, 56, 53, 59}
# After using intQ.pop_front()
# Size of the queue =9,
# intQ={60, 58, 55, 59, 53, 60, 56, 53, 59}
# After Using intQ.pop_back()
# Size of the oueue =8,
# intQ={60, 58, 55, 59, 53, 60, 56, 53}
# After using intQ.push_front(10)
# Size of the queue =9,
# intQ={10, 60, 58, 55, 59, 53, 60, 56, 53}
# After using intQ.back_front(17)
# Size of the queue =10,
# intQ={10, 60, 58, 55, 59, 53, 60, 56, 53, 17}
```

CODE-3 create a queue of classes in System Verilog.

```
class fruit;
string name;
function new (string name = "fruit");
    this.name = name;
endfunction
endclass

module top;
    fruit list[$];
    initial begin
        fruit f = new("apple");
        list.push_back(f);
        f = new("Banana");
        list.push_back(f);
        foreach(list[i])
            $display("list[%0d]=%s",i,list[i].name);
            $display("list=%p",list);
        end
    endmodule
```

```
# Top level modules:
# top
# End time: 00:08:09 on Sep 01,2023, Elapsed time: 0:00:01
# Errors: 0, Warnings: 2
# vsim top
# Start time: 00:07:03 on Sep 01,2023
# ** Note: (vsim-3812) Design is being optimized...
# ** Warning: class_queue.sv(14): (vopt-2244) Variable 'f' is i
# or remove the initialization in the declaration of variable.
# ** Note: (vsim-12126) Error and warning message counts have b
# Loading sv_std.std
# Loading work.class_queue_sv_unit(fast)
# Loading work.top(fast)
# list[0]=apple
# list[1]=Banana
# list='{@fruit@1, @fruit@2}
VSIM(paused)>
```