

Inputs:

- start_shift
 - no shift (1'b0)
 - shift (1'b1)
- start_read
 - no read (1'b0)
 - read (1'b1)
- shift_direc
 - no shift (2'b00)
 - left (2'b01)
 - right (2'b10)
 - down (2'b11)
- data_r
 - reg [7:0] data_r // 8 bit pixel value
- windowBuffer
 - reg [7:0] windowBuffer[0:8] // array with N-1 elements, each of M-1 bits
 - in the beginning : initialize this to all 0s

Outputs:

- read_done
 - no read (1'b0)
 - read done (1'b1)
- shift_done
 - no shift (1'b0)
 - shift done (1'b1)
- data
 - reg [7:0] tempWindowBuffer[0:8] // array with N+1 elements, each of M+1 bits
- windowBuffer
 - reg [7:0] tempWindowBuffer[0:8] // array with N-1 elements, each of M-1 bits

Algorithm:

Window Buffer Diagram:

0	3	6
1	4	7
2	5	8

★ **NOTE:** The numbers indicate the order in which window block will be inserting pixels into the window buffer. *** So make sure that we read pixels and provide them as input to the window block in this order!!! ***

1) `reg [7:0] tempWindowBuffer[0:8] = windowBuffer`

2) CHECK `start_shift`

3) IF `start_shift` high:

 a) CHECK `shift_dirac`

 i) IF `shift_dirac == 2'b01` // shift left

 (1) `tempWindowBuffer [0] = tempWindowBuffer [3]`

 (2) `tempWindowBuffer [1] = tempWindowBuffer [4]`

 (3) `tempWindowBuffer [2] = tempWindowBuffer [5]`

 (4) `tempWindowBuffer [3] = tempWindowBuffer [6]`

 (5) `tempWindowBuffer [4] = tempWindowBuffer [7]`

 (6) `tempWindowBuffer [5] = tempWindowBuffer [8]`

 (7) `tempWindowBuffer [6] = 0`

 (8) `tempWindowBuffer [7] = 0`

 (9) `tempWindowBuffer [8] = 0`

 (10) `shift_done = 1`

 ii) IF `shift_dirac == 2'b10` // shift right

 (1) `tempWindowBuffer [8] = tempWindowBuffer [5]`

 (2) `tempWindowBuffer [7] = tempWindowBuffer [4]`

 (3) `tempWindowBuffer [6] = tempWindowBuffer [3]`

 (4) `tempWindowBuffer [5] = tempWindowBuffer [2]`

 (5) `tempWindowBuffer [4] = tempWindowBuffer [1]`

 (6) `tempWindowBuffer [3] = tempWindowBuffer [0]`

```
(7) tempWindowBuffer [0] = 0
(8) tempWindowBuffer [1] = 0
(9) tempWindowBuffer [2] = 0
```

```
(10) shift_done = 1
```

```
iii) IF shift_direct == 2'b11 // shift down
```

```
(1) tempWindowBuffer [2] = tempWindowBuffer [1]
(2) tempWindowBuffer [5] = tempWindowBuffer [4]
(3) tempWindowBuffer [8] = tempWindowBuffer [7]
(4) tempWindowBuffer [1] = tempWindowBuffer [0]
(5) tempWindowBuffer [4] = tempWindowBuffer [3]
(6) tempWindowBuffer [7] = tempWindowBuffer [6]
(7) tempWindowBuffer [0] = 0
(8) tempWindowBuffer [3] = 0
(9) tempWindowBuffer [6] = 0
```

```
(10) shift_done = 1
```

4) ELSE IF start_read high:

a) Find first empty element in tempWindowBuffer

```
i) i = 0; notFound = 1; firstEmpty = 0;
   while ( (i < 9) & notFound )
   {
       if tempWindowBuffer[i] == 0
       {
           firstEmpty = i;
           notFound = 0;
       }
       i = i + 1;
   }
```

b) Insert data_r

```
i) tempWindowBuffer[firstEmpty] = data_r
ii) read_done = 1
```

5) Assign Outputs

a) assign windowBuffer = tempWindowBuffer
b) assign data = tempWindowBuffer