**Lab 12: Turing Machine Design and Development**

**Purpose: Learn and implement a Turing Machine**

A Turing machine is defined on *S*={*0, 1*, *halt*}, and *I*={*0, 1*}

Note: add a ‘halt’ state to indicate that the machine halts

The set of quintuples is listed below:

(0,1,1,0,R)

(0,0,1,1,R)

1. What is the destination register?

tape\_reg with length 8, and with initial value 8’b00111000

1. Do we need any source register(s)? Any other registers do we need?

Source register is the same as the destination register: tape\_reg

turing\_state for indicating the Turing machine’s state

done\_reg to indicate that the machine halts

1. How do we indicate where the reading head is?

Use an index for indicating where the reading head is

1. Describe the Turning machine operations using RT operations notation

Note: the ‘if’ statement, and ‘loop’ statements are allowed

tape\_reg ← 8’b00111000;

index ← 5;

turing\_state←0;

done\_reg←0;

while (turing\_state != halt)

{ if (turing\_state == 0)&&(tape\_reg[index]==’1’)

{ index←index-1;}

if (turing\_state == 0)&&(tape\_reg[index]==’0’)

{ tape\_reg[index]=’1’;

index←index-1;

turing\_state←1;

}

if (turing\_state==1)

{ turing\_state=halt;

done\_reg←1;}

}

Discussions: Can we reduce the number of states? Combining state ‘1’ and ‘halt’ together?

tape\_reg ← 8’b00111000;

index ← 5;

turing\_state←0;

done\_reg←0;

while (turing\_state != 1)

{ if (turing\_state == 0)&&(tape\_reg[index]==’1’)

{ index←index-1;}

if (turing\_state == 0)&&(tape\_reg[index]==’0’)

{ tape\_reg[index]=’1’;

index←index-1;

turing\_state←1;

done\_reg←1;

}

}

1. **Convert the RT operations into ASMD**

*s1, s2, s3* are the FSM’s states

tape\_reg ← 8’b00111000;

index ← 5;

turing\_state←0;

done\_reg←0;

***s1***

reset

no

Ready?

yes

yes

***s2***

***s3***

halt

turing state!= 1?

no

turing state == 0

&& tape\_reg[index]==1b’0?

no

yes

tape\_reg[index] ← 1b’1;

turing\_state←1;

done\_reg←1;

index ← index-1;

1. Decide inputs and outputs
2. Develop the Verilog code follow the template