

VIETNAM NATIONAL UNIVERSITY, HO CHI MINH CITY

UNIVERSITY OF INFORMATION TECHNOLOGY

FACULTY OF COMPUTER ENGINEERING



**FINAL TERM HOMEWORK REPORT**

**Mentor: Lâm Đức Khải**

**Student: Trần Nguyễn Anh Khoa - 21521004**

# TABLE OF CONTENTS

<b>TASK 1: Square-root approximation design with Left-edge algorithm .....</b>	<b>11</b>
1.1.    Overall Design .....	12
1.2.    Datapath Design.....	12
1.2.1.    Mux_3to1_8bit .....	13
1.2.2.    Mux_5to1_8bit .....	13
1.2.3.    Register_8bit.....	13
1.2.4.    Absolute_8bit.....	13
1.2.5.    Min_8bit .....	14
1.2.6.    Max_8bit.....	14
1.2.7.    Full_Adder_8bit.....	14
1.2.8.    Subtractor_8bit .....	14
1.2.9.    Shift-1-Right_8bit.....	15
1.2.10.    Shift-3-Right_8bit.....	15
1.3.    Controller Design.....	15
1.3.1.    Next_state .....	17
1.3.2.    Present_state .....	17
1.3.3.    Logic_output.....	18
1.4.    Waveform simulation .....	18
<b>TASK 2: SRA design with Graph-partitioning algorithm .....</b>	<b>19</b>
2.1.    Overall Design .....	20
2.2.    Datapath Design.....	20
2.2.1.    Mux_3to1_8bit .....	21
2.2.2.    Mux_6to1_8bit .....	21
2.2.3.    Register_8bit.....	21
2.2.4.    Absolute_8bit.....	22
2.2.5.    Min_8bit .....	22
2.2.6.    Max_8bit.....	22
2.2.7.    Full_Adder_8bit.....	22
2.2.8.    Subtractor_8bit .....	23
2.2.9.    Shift-1-Right_8bit.....	23
2.2.10.    Shift-3-Right_8bit.....	23
2.3.    Controller Design.....	24

2.3.1.	Next_state .....	25
2.3.2.	Present_state .....	25
2.3.3.	Logic_output.....	26
2.4.	Waveform simulation .....	26
<b>TASK 3: SRA design with prioritized partitioning</b>	.....	<b>27</b>
Basic theory:	.....	27
3.1.	Overall Design .....	27
3.2.	Datapath Design.....	27
3.2.1.	Mux_3to1_8bit .....	28
3.2.2.	Register_8bit.....	28
3.2.3.	abs_max .....	28
3.2.4.	abs_min_add_sub .....	29
3.2.5.	Full_Adder_8bit.....	29
3.2.6.	Shift-1-Right_8bit.....	30
3.2.7.	Shift-3-Right_8bit.....	30
3.3.	Controller Design.....	31
3.3.1.	Next_state .....	32
3.3.2.	Present_state .....	32
3.3.3.	Logic_output.....	33
3.4.	Waveform simulation .....	33
<b>TASK 4: SRA design with connection merging</b>	.....	<b>34</b>
4.1.	Overall Design .....	34
4.2.	Datapath Design.....	34
4.2.1.	Register_8bit.....	35
4.2.2.	abs_max .....	35
4.2.3.	abs_min_add_sub .....	36
4.2.4.	Full_Adder_8bit.....	36
4.2.5.	Shift-1-Right_8bit.....	37
4.2.6.	Shift-3-Right_8bit.....	37
4.3.	Controller Design.....	38
4.3.1.	Next_state .....	39
4.3.2.	Present_state .....	39
4.3.3.	Logic_output.....	40

4.4.	Waveform simulation .....	40
<b>Task 5: SRA design with Functional-unit pipelining.....</b>		<b>41</b>
Basic theory: .....		41
5.1.	Overall Design .....	41
5.2.	Datapath Design.....	42
5.2.1.	Register_8bit.....	42
5.2.2.	AU-2-stage.....	43
5.2.3.	Full_Adder_8bit.....	44
5.2.4.	Shift-1-Right_8bit.....	44
5.2.5.	Shift-3-Right_8bit.....	44
5.3.	Controller Design.....	45
5.3.1.	Next_state .....	46
5.3.2.	Present_state .....	46
5.3.3.	Logic_output.....	47
5.4.	Waveform simulation .....	48
<b>Task 6: SRA design with Datapath pipelining.....</b>		<b>49</b>
Basic theory: .....		49
6.1.	Overall Design .....	49
6.2.	Datapath Design.....	49
6.2.1.	Register_8bit.....	50
6.2.2.	abs_min_max .....	50
6.2.3.	add_sub_max .....	51
6.2.4.	Full_Adder_8bit.....	51
6.2.5.	Shift-1-Right_8bit.....	52
6.2.6.	Shift-3-Right_8bit.....	52
6.3.	Controller Design.....	53
6.3.1.	Next_state .....	54
6.3.2.	Present_state .....	54
6.3.3.	Logic_output.....	55
6.4.	Waveform simulation .....	57
<b>Task 7: Datapath pipelining with Functional Unit Pipeline.....</b>		<b>59</b>
Basic theory: .....		59
7.1.	Overall Design .....	59

7.2.	Datapath Design.....	60
7.2.1.	Register_8bit.....	60
7.2.2.	AU1_2-stage.....	61
7.2.2.	AU2_2-stage.....	61
7.2.3.	Full_Adder_8bit.....	61
7.2.4.	Shift-1-Right_8bit.....	62
7.2.5.	Shift-3-Right_8bit.....	62
7.3.	Controller Design.....	63
7.3.1.	Next_state .....	64
7.3.2.	Present_state .....	65
7.3.3.	Logic_output.....	65
7.4.	Waveform simulation .....	69

# LIST OF FIGURES

Figure 1: SRA algorithm .....	11
Figure 2: Left-edge algorithm.....	11
Figure 3: Task 1 overall symbol .....	12
Figure 4: Task 1 overall circuit .....	12
Figure 5: Task 1 datapath symbol.....	12
Figure 6: Task 1 datapath circuit .....	12
Figure 7: Task 1 Mux_3to1_8bit symbol .....	13
Figure 8: Task 1 Mux_3to1_8bit circuit.....	13
Figure 9: Task 1 Mux_5to1_8bit symbol .....	13
Figure 10: Task 1 Mux_5to1_8bit circuit.....	13
Figure 11: Task 1 Register_8bit symbol.....	13
Figure 12: Task 1 Register_8bit circuit .....	13
Figure 13: Task 1 Absolute_8bit symbol.....	14
Figure 14: Task 1 Absolute_8bit circuit.....	14
Figure 15: Task 1 Min_8bit symbol .....	14
Figure 16: Task 1 Min_8bit circuit .....	14
Figure 17: Task 1 Max_8bit symbol.....	14
Figure 18: Task 1 Max_8bit circuit .....	14
Figure 19: Task 1 Full_Adder_8bit symbol.....	14
Figure 20: Task 1 Full_Adder_8bit circuit .....	14
Figure 21: Task 1 Full_Subtractor_8bit symbol .....	15
Figure 22: Task 1 Full_Subtractor_8bit circuit .....	15
Figure 23: Task 1 Shift_1_Right_8bit symbol .....	15
Figure 24: Task 1 Shift_1_Right_8bit cirult.....	15
Figure 25: Task 1 Shift_3_Right_8bit symbol .....	15
Figure 26: Task 1 Shift_3_Right_8bit circuit .....	15
Figure 27: Task 1 Controller symbol.....	16
Figure 28: Task 1 Controller circuit .....	16
Figure 29: Task 1 state diagram.....	16
Figure 30: Task 1 next state symbol .....	17
Figure 31: Task 1 next state circuit.....	17
Figure 32: Task 1 present state symbol .....	17
Figure 33: Task 1 present state circuit.....	17
Figure 34: Task 1 logic output symbol .....	18
Figure 35: Task 1 logic output circuit.....	18
Figure 36: Task 1 Waveform.....	18
Figure 37: Graph Partintioning algorithm.....	19
Figure 38: Initial compatibility graph.....	19
Figure 39: Compatibility graph after merging t3, t5 and t6.....	19
Figure 40:Compatibility graph after merging t1, x and t7 .....	19
Figure 41: Compatibility graph after merging t2 and y .....	19
Figure 42: Final compatibility graph .....	19

Figure 43: Task 2 overall design symbol.....	20
Figure 44: Task 2 overall design circuit .....	20
Figure 45: Task 2 Datapath symbol.....	20
Figure 46: Task 2 Datapath circuit .....	20
Figure 47: Task 2 Mux_3to1_8bit symbol .....	21
Figure 48: Task 2 Mux_3to1_8bit circuit.....	21
Figure 49: Task 2 Mux_6to1_8bit symbol .....	21
Figure 50: Task 2 Mux_6to1_8bit circuit.....	21
Figure 51: Task 2 Register_8bit symbol.....	21
Figure 52: Task 2 Register_8bit circuit .....	21
Figure 53: Task 2 Absolute_8bit symbol.....	22
Figure 54: Task 2 Absolute_8bit circuit .....	22
Figure 55: Task 2 Min_8bit symbol .....	22
Figure 56: Task 2 Min_8bit circuit .....	22
Figure 57: Task 2 Max_8bit symbol.....	22
Figure 58: Task 2 Max_8bit circuit .....	22
Figure 59: Task 2 Full_Adder_8bit symbol.....	22
Figure 60: Task 2 Full_Adder_8bit circuit .....	22
Figure 61: Task 2 Full_Subtractor_8bit symbol .....	23
Figure 62: Task 2 Full_Subtractor_8bit circuit .....	23
Figure 63: Task 2 Shift_1_Right_8bit symbol .....	23
Figure 64: Task 2 Shift_1_Right_8bit circuit.....	23
Figure 65: Task 2 Shift_3_Right_8bit symbol .....	23
Figure 66: Task 2 Shift_3_Right_8bit circuit .....	23
Figure 67: Task 2 Controller symbol.....	24
Figure 68: Task 2 Controller circuit .....	24
Figure 69: Task 2 state diagram.....	24
Figure 70: Task 2 Next_State symbol .....	25
Figure 71: Task 2 Next_state circuit.....	25
Figure 72: Task 2 Present_State symbol .....	25
Figure 73: Task 2 Present_State circuit.....	25
Figure 74: Task 2 Logic_Output symbol.....	26
Figure 75: Task 2 Logic_output circuit .....	26
Figure 76: Task 2 Waveform.....	26
Figure 77: Task 3 Overall design symbol .....	27
Figure 78: Task 3 overall design circuit .....	27
Figure 79: Task 3 datapth symbol .....	27
Figure 80: Task 3 datapath circuit .....	27
Figure 81: Task 3 Mux_3to1_8bit symbol .....	28
Figure 82: Task 3 Mux_3to1_8bit circuit.....	28
Figure 83: Task 3 Register_8bit symbol.....	28
Figure 84: Task 3 Register_8bit circuit .....	28
Figure 85: Task 3 Abs_max symbol .....	28
Figure 86: Task 3 Abs_max circuit .....	28

Figure 87: Task 3 Abs_Min_Add_Sub symbol .....	29
Figure 88: Task 3 Abs_Min_Add_Sub circuit.....	29
Figure 89: Task 3 Full_Adder_8bit symbol.....	29
Figure 90: Task 3 Full_Adder_8bit circuit .....	29
Figure 91: Task 3 Shift_1_Right_8bit symbol .....	30
Figure 92: Task 3 Shift_1_Right_8bit circuit.....	30
Figure 93: Task 3 Shift_3_Right_8bit symbol .....	30
Figure 94: Task 3 Shift_3_Right_8bit circuit.....	30
Figure 95: Task 3 Controller symbol.....	31
Figure 96: Task 3 Controller circuit .....	31
Figure 97: Task 3 state diagram.....	31
Figure 98: Task 3 Next_State symbol .....	32
Figure 99: Task 3 Next_State circuit.....	32
Figure 100: Task 3 Present_State symbol .....	32
Figure 101: Task 3 Present_State circuit.....	32
Figure 102: Task 3 Logic_Output symbol.....	33
Figure 103: Task 3 Logic_Output circuit .....	33
Figure 104: Task 3 Waveform.....	33
Figure 107: Task 4 overall symbol .....	34
Figure 108: Task 4 overall circuit.....	34
Figure 109: Task 4 datapath symbol.....	34
Figure 110: Task 4 datapath circuit .....	35
Figure 111: Task 4 Register_8bit symbol .....	35
Figure 112: Task 4 Register_8bit circuit .....	35
Figure 113: Task 4 Abs_Max symbol .....	35
Figure 114: Task 4 Abs_Max circuit.....	35
Figure 115: Task 4 Abs_Min_Add_Sub symbol .....	36
Figure 116: Task 4 Abs_Min_Add_Sub circuit.....	36
Figure 117: Task 4 Full_Adder_8bit symbol.....	36
Figure 118: Task 4 Full_Adder_8bit circuit .....	36
Figure 119: Task 4 Shift_1_Right_8bit symbol .....	37
Figure 120: Task 4 Shift_1_Right_8bit circuit .....	37
Figure 121: Task 4 Shift_3_Right_8bit symbol .....	37
Figure 122: Task 4 Shift_3_Right_8bit circuit .....	37
Figure 123: Task 4 controller symbol.....	38
Figure 124: Task 4 controller circuit .....	38
Figure 125: Task 4 state diagram.....	38
Figure 126: Task 4 Next_State symbol .....	39
Figure 127: Task 4 Next_State circuit.....	39
Figure 128: Task 4 Present_State symbol .....	39
Figure 129: Task 4 Present_State circuit.....	39
Figure 130: Task 4 Logic_Output symbol.....	40
Figure 131: Task 4 Logic_Output circuit .....	40
Figure 132: Task 4 Waveform.....	40

Figure 133: Task 5 timing diagram .....	41
Figure 134: Task 5 overall symbol .....	41
Figure 135: Task 5 overall circuit.....	41
Figure 136: Task 5 datapath symbol.....	42
Figure 137: Task 5 datapath circuit .....	42
Figure 138: Task 5 Resgister_8bit symbol .....	42
Figure 139: Task 5 Resgister_8bit circuit.....	42
Figure 140: Task 5 AU_2_Stage symbol.....	43
Figure 141: Task 5 AU_2_Stage circuit .....	43
Figure 142: Task 5 Full_Adder_8bit symbol.....	44
Figure 143: Task 5 Full_Adder_8bit circuit .....	44
Figure 144: Task 5 Shift_1_Right_8bit symbol .....	44
Figure 145: Task 5 Shift_1_Right_8bit circuit .....	44
Figure 146: Task 5 Shift_3_Right_8bit symbol .....	44
Figure 147: Task 5 Shift_3_Right_8bit circui .....	44
Figure 148: Task 5 controller symbol.....	45
Figure 149: Task 5 controller circuit .....	45
Figure 150: Task 5 state diagram.....	45
Figure 151: Task 5 Next_State symbol .....	46
Figure 152: Task 5 Next_State circuit.....	46
Figure 153: Task 5 Present_State symbol .....	46
Figure 154: Task 5 Present_State circuit.....	46
Figure 155: Task 5 Logic_Output symbol.....	47
Figure 156: Task 5 Logic_Output circuit .....	47
Figure 157: Task 5 waveform.....	48
Figure 158: Task 6 overall symbol .....	49
Figure 159: Task 6 overall circuit.....	49
Figure 160: Task 6 datapath symbol.....	49
Figure 161: Task 6 datapath circuit .....	49
Figure 162: Task 6 Register_8bit symbol.....	50
Figure 163: Task 6 Register_8bit circuit .....	50
Figure 164: Task 6 Abs_Min_Max symbol.....	50
Figure 165: Task 6 Abs_Min_Max circuit .....	50
Figure 166: Task 6 Abs_Sub_Max symbol .....	51
Figure 167: Task 6 Abs_Sub_Max circuit.....	51
Figure 168: Task 6 Full_Adder_8bit symbol.....	51
Figure 169: Task 6 Full_Adder_8bit circuit .....	51
Figure 170: Task 6 Shift_1_Right_8bit symbol .....	52
Figure 171: Task 6 Shift_1_Right_8bit circuit .....	52
Figure 172: Task 6 Shift_3_Right_8bit symbol .....	52
Figure 173: Task 6 Shift_3_Right_8bit circuit.....	52
Figure 174: Task 6 controller symbol.....	53
Figure 175: Task 6 controller circuit .....	53
Figure 176: Task 6 state diagram.....	53

Figure 177: Task 6 Next_State symbol .....	54
Figure 178: Task 6 Next_State circuit.....	54
Figure 179: Task 6 Present_State symbol .....	54
Figure 180: Task 6 Present_State circuit.....	54
Figure 181: Task 6 Logic_Output symbol.....	55
Figure 182: Task 6 Logic_Output circuit .....	56
Figure 183: Task 6 waveform.....	58
Figure 184: Task 7 overall symbol .....	59
Figure 185: Task 7 overall circuit.....	59
Figure 186: Task 7 datapath symbol.....	60
Figure 187: Task 7 datapath circuit .....	60
Figure 188: Task 7 Register_8bit symbol.....	60
Figure 189: Task 7 Register_8bit circuit .....	60
Figure 190: Task 7 AU1_2_Stage symbol.....	61
Figure 191: Task 7 AU1_2_Stage circuit .....	61
Figure 192: Task 7 AU2_2_Stage symbol.....	61
Figure 193: Task 7 AU2_2_Stage circuit .....	61
Figure 194: Task 7 Full_Adder_8bit symbol.....	61
Figure 195: Task 7 Full_Adder_8bit circuit .....	61
Figure 196: Task 7 Shift_1_Right_8bit symbol .....	62
Figure 197: Task 7 Shift_1_Right_8bit circuit .....	62
Figure 198: Task 7 Shift_3_Right_8bit symbol .....	62
Figure 199: Task 7 Shift_3_Right_8bit circuit .....	62
Figure 200: Task 7 controller symbol.....	63
Figure 201: Task 7 controller circuit .....	63
Figure 202: Task 7 state diagram.....	64
Figure 203: Task 7 Next_State symbol .....	64
Figure 204: Task 7 Next_State circuit.....	64
Figure 205: Task 7 Present_State symbol .....	65
Figure 206: Task 7 Present_State circuit .....	65
Figure 207: Task 7 Logic_Output symbol.....	65
Figure 208: Task 7 Logic_Output circuit .....	68
Figure 209: Task 7 waveform.....	69

## TASK 1: Square-root approximation design with Left-edge algorithm

Design the square-root approximation (SRA) of two signed integers,  $a$  and  $b$ , by the following formula:

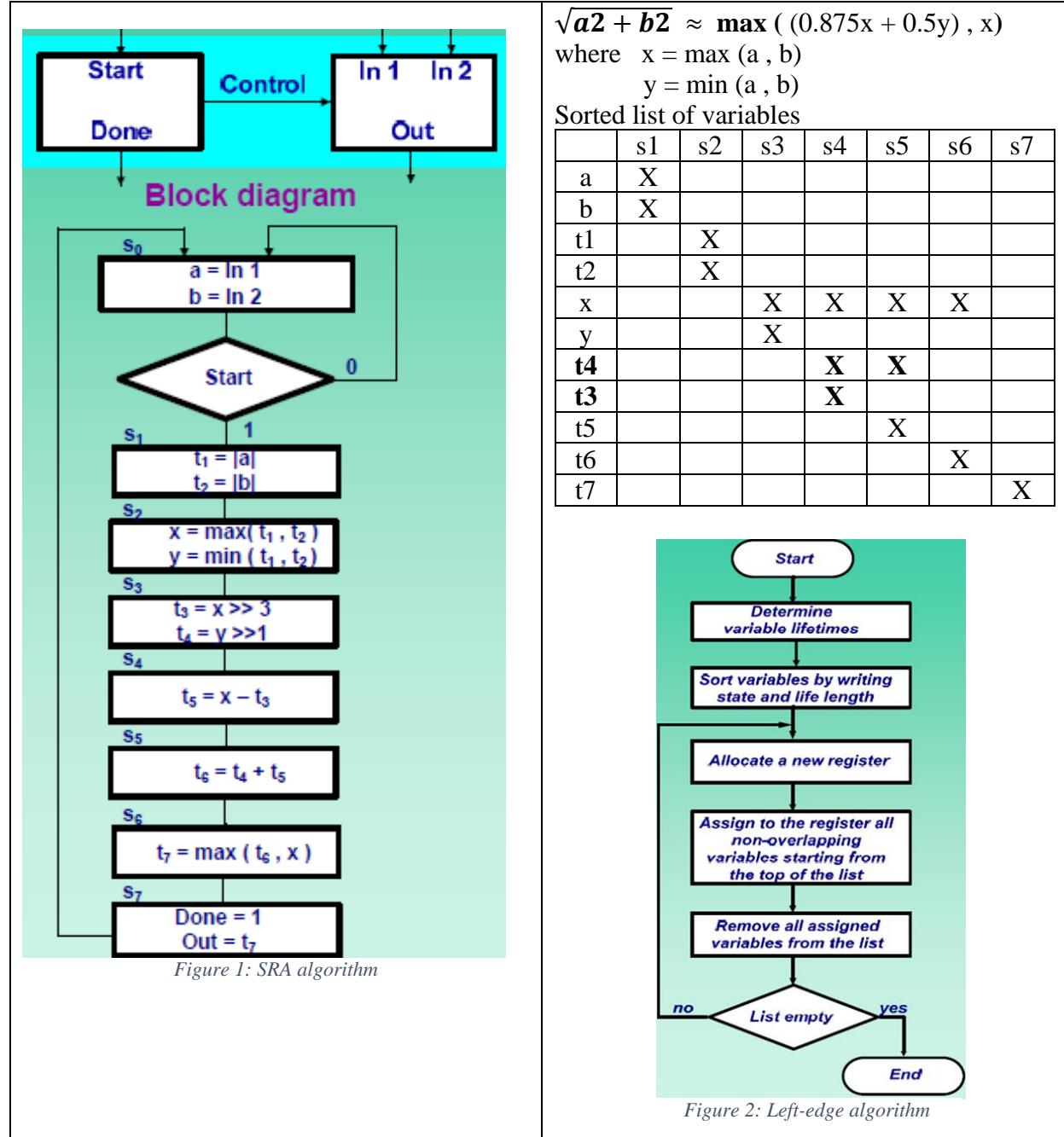


Figure 1: SRA algorithm

Priority order for 2 variables have the **same write state**:

1. The variable with the longer lifetime
2. The priority will be assigned at random

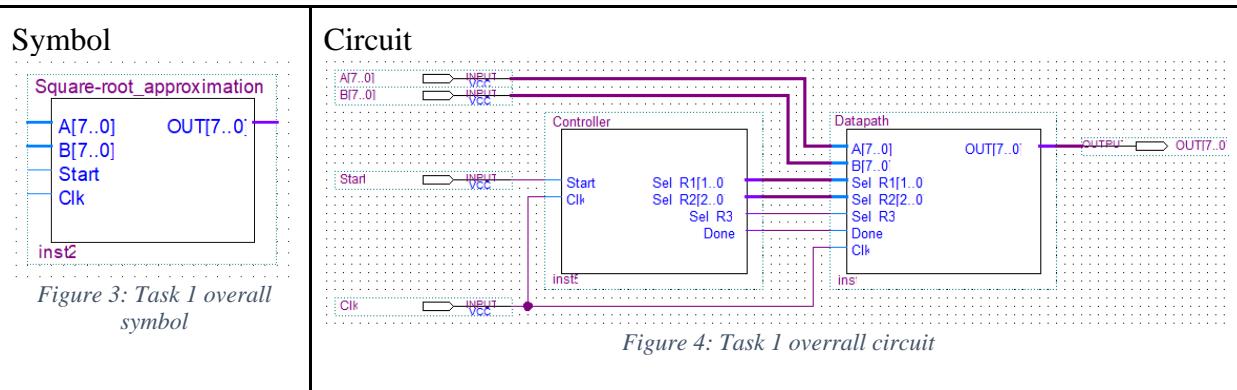
### Register assignments

$$R1 = \{a, t1, x, t7\}$$

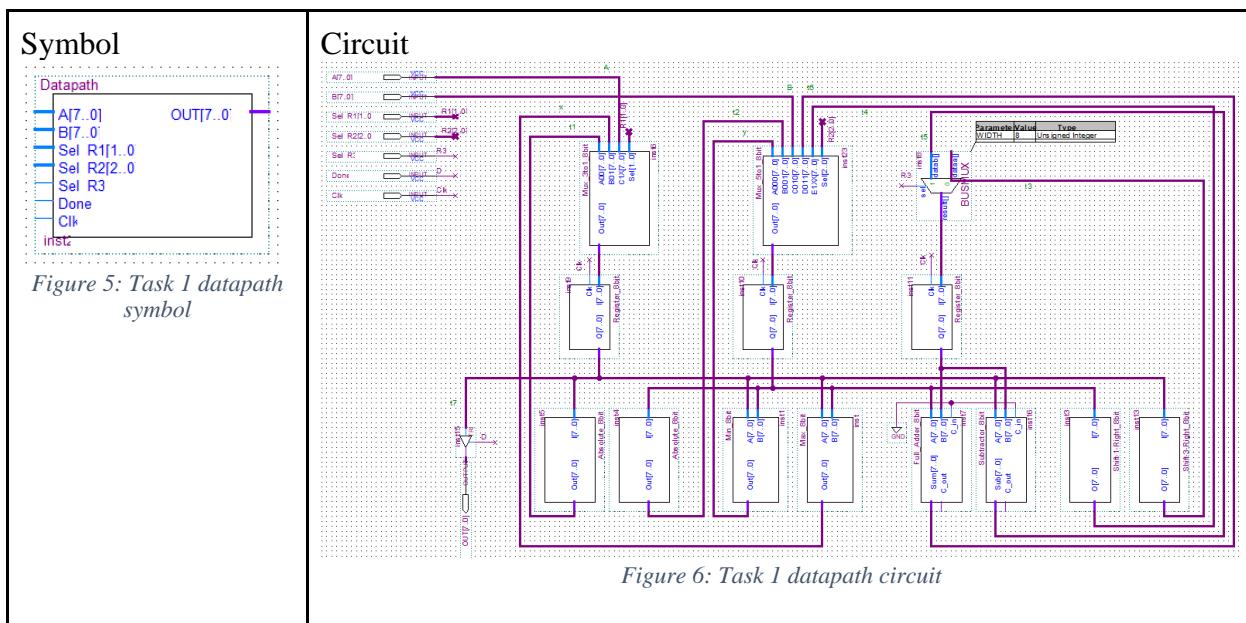
$$R2 = \{b, t2, y, t4, t6\}$$

$$R3 = \{t3, t5\}$$

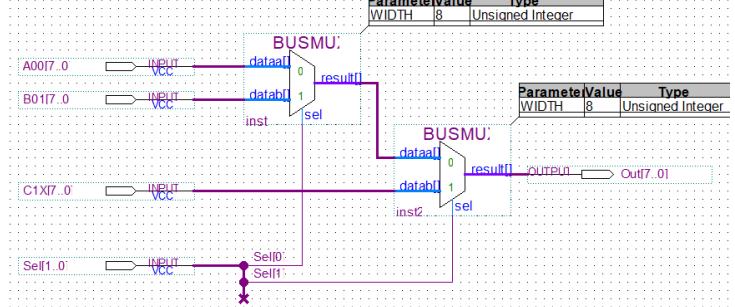
## 1.1. Overall Design



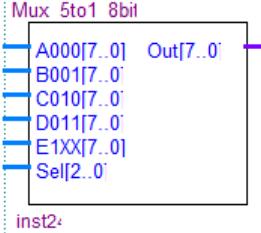
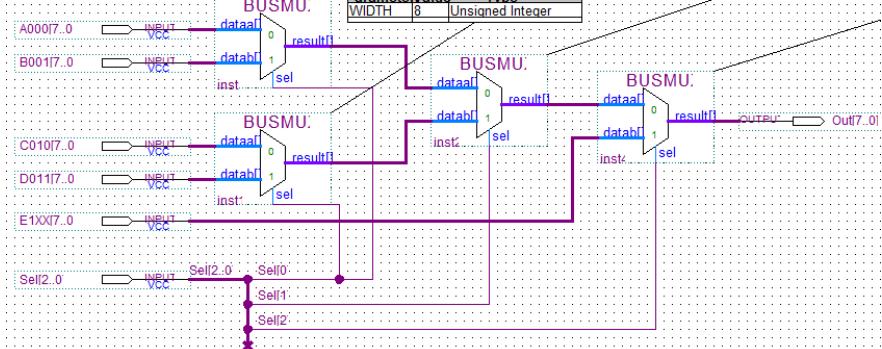
## 1.2. Datapath Design



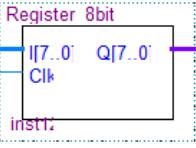
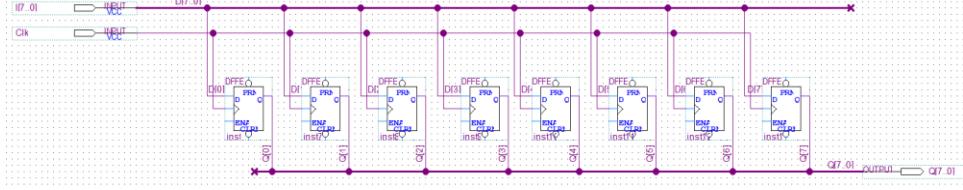
### 1.2.1. Mux\_3to1\_8bit

Symbol	Circuit
 <p>Figure 7: Task 1 Mux_3to1_8bit symbol</p>	 <p>Figure 8: Task 1 Mux_3to1_8bit circuit</p>

### 1.2.2. Mux\_5to1\_8bit

Symbol	Circuit
 <p>Figure 9: Task 1 Mux_5to1_8bit symbol</p>	 <p>Figure 10: Task 1 Mux_5to1_8bit circuit</p>

### 1.2.3. Register\_8bit

Symbol	Circuit
 <p>Figure 11: Task 1 Register_8bit symbol</p>	 <p>Figure 12: Task 1 Register_8bit circuit</p>

### 1.2.4. Absolute\_8bit

Symbol	Circuit
--------	---------

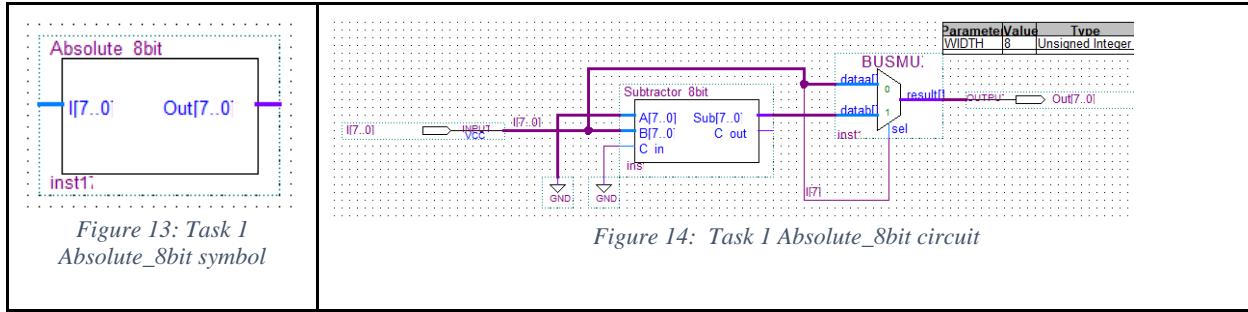


Figure 13: Task 1  
Absolute\_8bit symbol

Figure 14: Task 1 Absolute\_8bit circuit

### 1.2.5. Min\_8bit

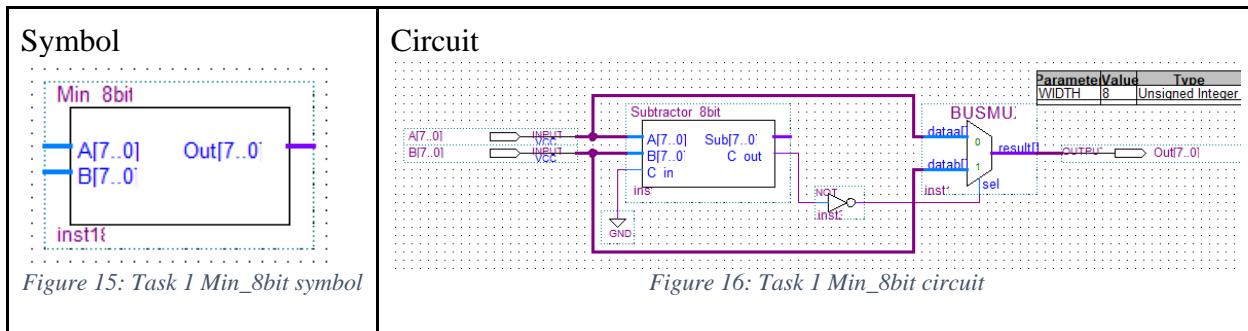


Figure 15: Task 1 Min\_8bit symbol

Figure 16: Task 1 Min\_8bit circuit

### 1.2.6. Max\_8bit

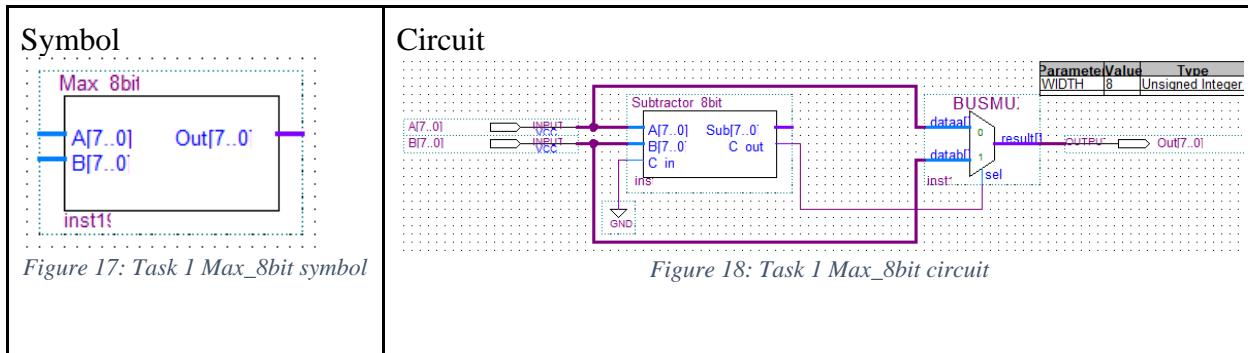


Figure 17: Task 1 Max\_8bit symbol

Figure 18: Task 1 Max\_8bit circuit

### 1.2.7. Full\_Adder\_8bit

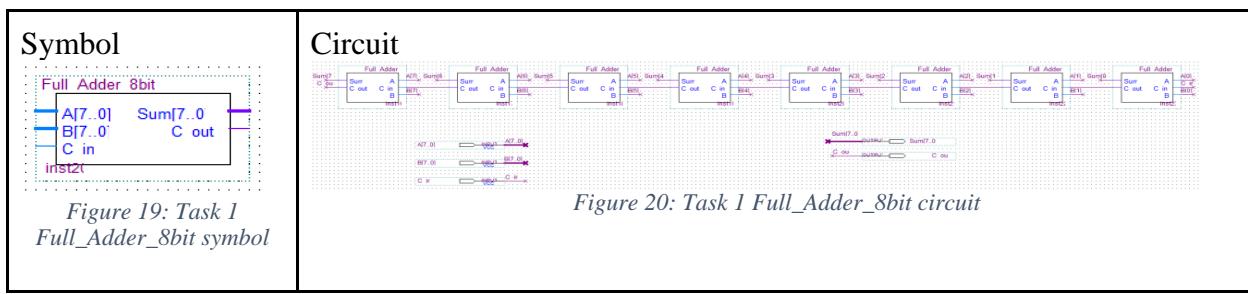
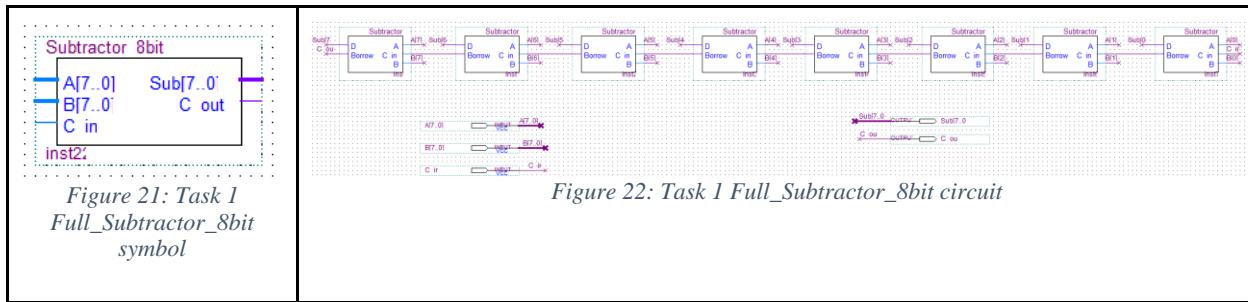


Figure 19: Task 1  
Full\_Adder\_8bit symbol

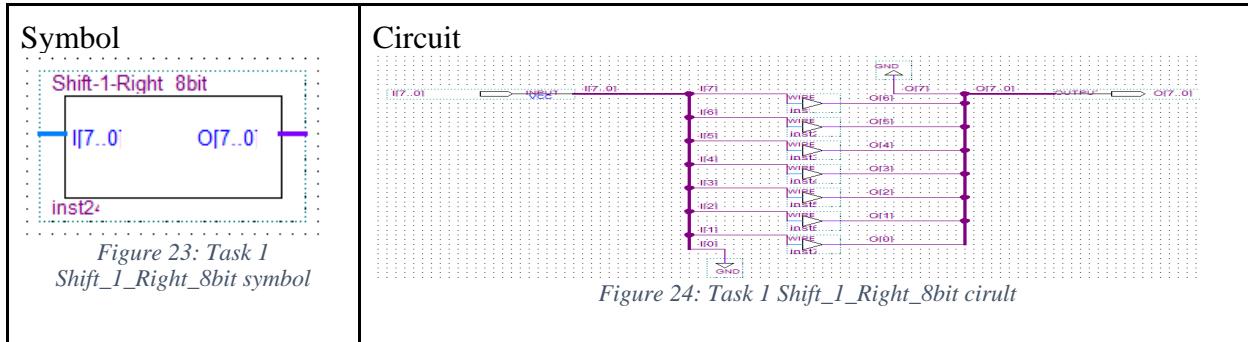
Figure 20: Task 1 Full\_Adder\_8bit circuit

### 1.2.8. Subtractor\_8bit

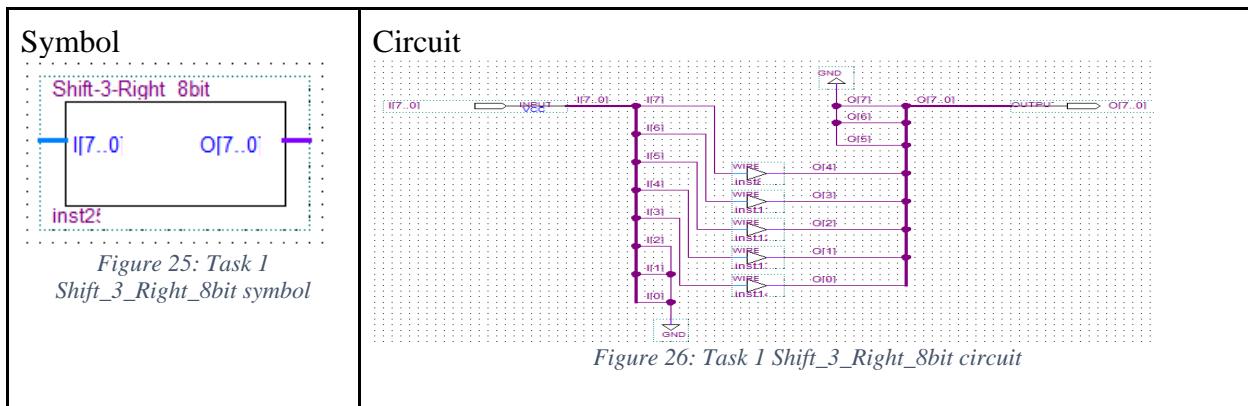




### 1.2.9. Shift-1-Right\_8bit

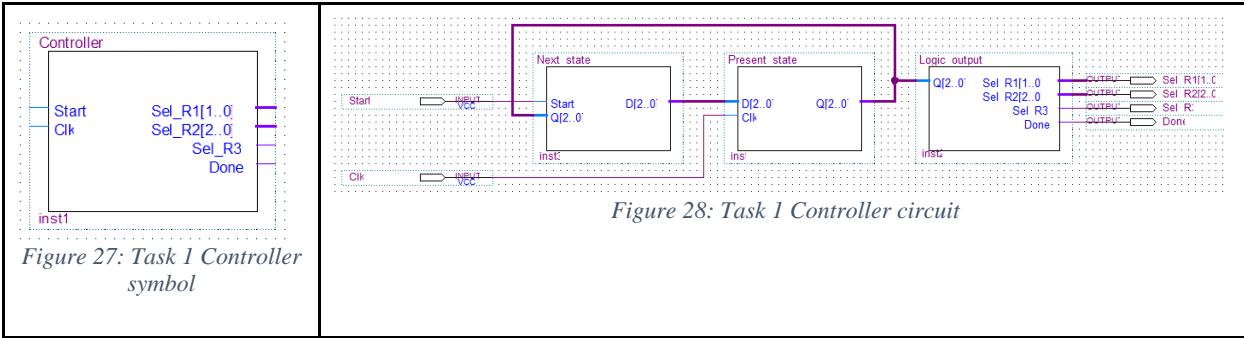


### 1.2.10. Shift-3-Right\_8bit



## 1.3. Controller Design

Symbol	Circuit
--------	---------



State Table

	Q2Q1Q0	Start	Q2+Q1+Q0+	Sel_R1	Sel_R2	Sel_R3	Done
s0	000	0	000	1X	010	0	0
	000	1	001	1X	010	0	0
s1	001	0	010	00	001	0	0
	001	1	010	00	001	0	0
s2	010	0	011	01	000	0	0
	010	1	011	01	000	0	0
s3	011	0	100	01	1XX	0	0
	011	1	100	01	1XX	0	0
s4	100	0	101	01	000	1	0
	100	1	101	01	000	1	0
s5	101	0	110	01	011	0	0
	101	1	110	01	011	0	0
s6	110	0	111	01	000	0	0
	110	1	111	01	000	0	0
s7	111	0	111	01	000	0	1
	111	1	000	01	000	0	1

State Diagram

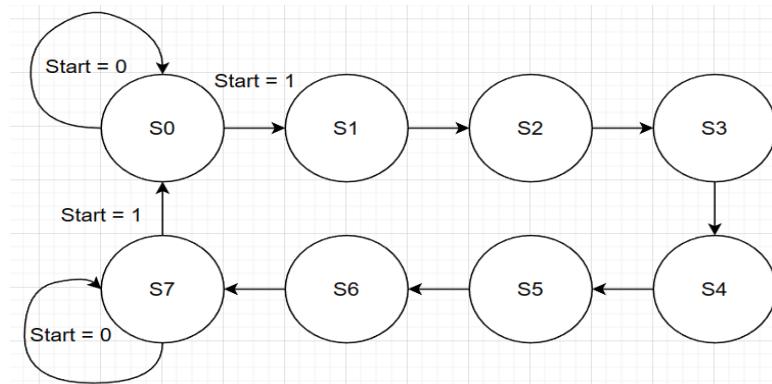
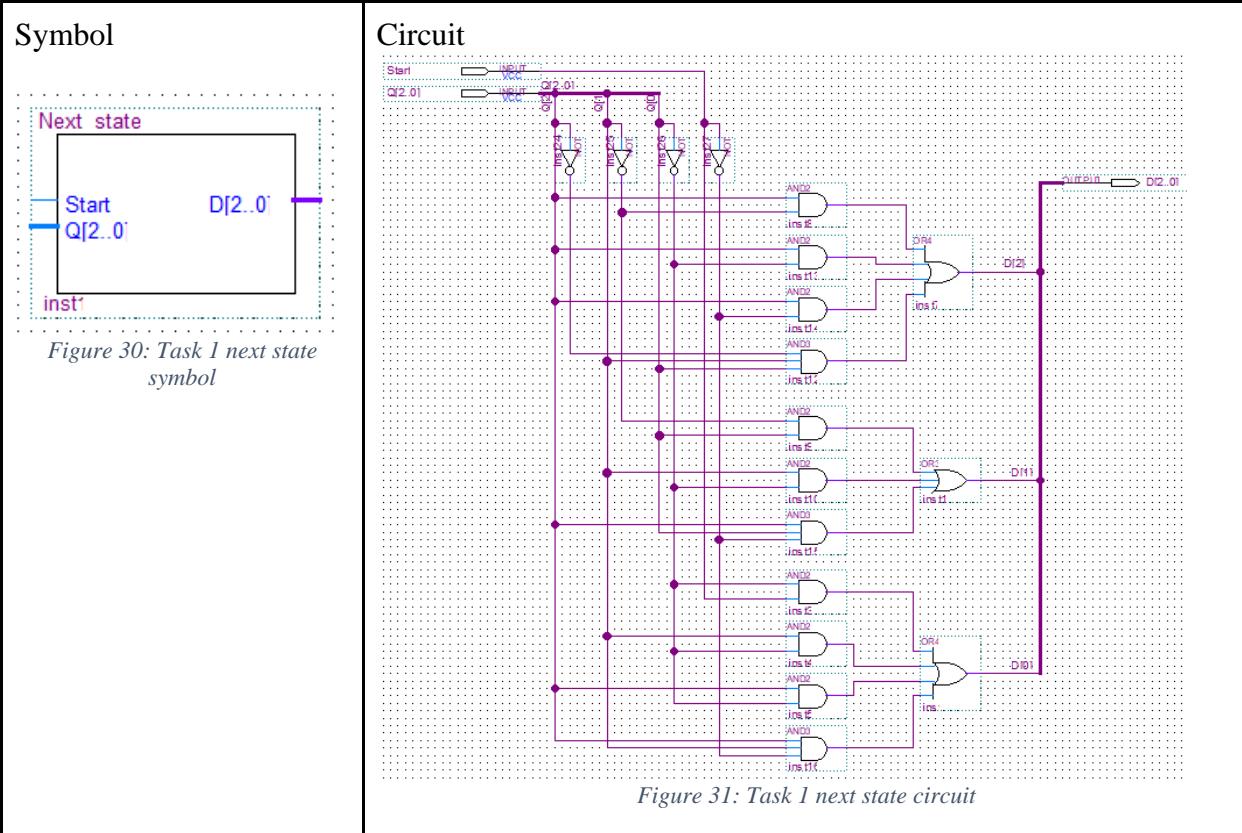


Figure 29: Task 1 state diagram

### 1.3.1. Next\_state



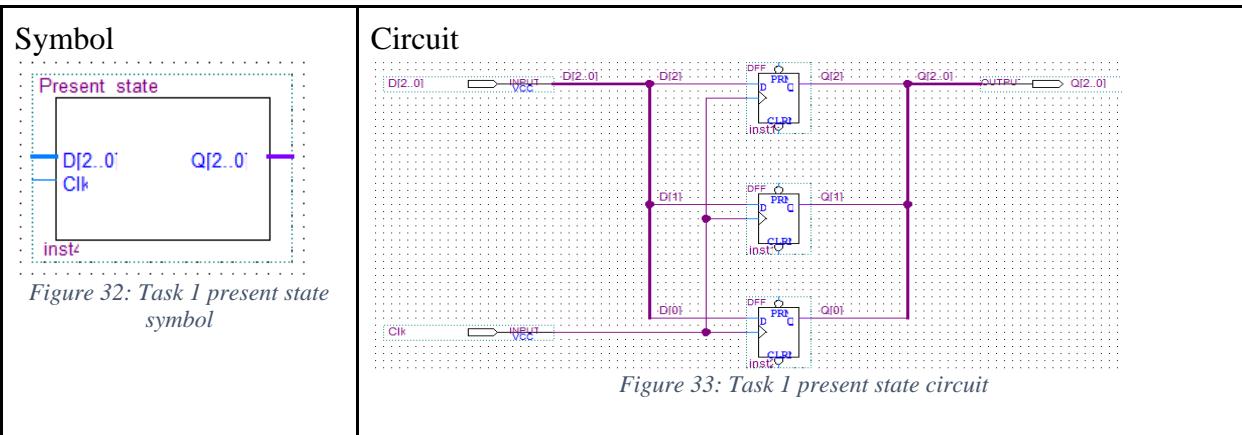
Next\_state equation

$$Q2+ = Q2.Q1' + Q2.Q0' + Q2.Start' + Q2'.Q1.Q0$$

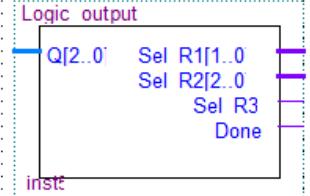
$$Q1+ = Q1'.Q0 + Q1.Q0' + Q2.Q0.Start'$$

$$Q0+ = Q0'.Start + Q1.Q0' + Q2.Q0' + Q2.Q1.Start'$$

### 1.3.2. Present\_state



### 1.3.3. Logic\_output

Symbol	Circuit
 <p>Figure 34: Task 1 logic output symbol</p>	<p>Figure 35: Task 1 logic output circuit</p>

Logic_output equation	
$Sel\_R1[1] = Q2'.Q1'.Q0'$	$Sel\_R2[0] = Q1'.Q0$
$Sel\_R1[0] = Q1 + Q2$	$Sel\_R3 = Q2.Q1'.Q0'$
$Sel\_R2[2] = Q2'.Q1.Q0$	$Done = Q2.Q1.Q0$
$Sel\_R2[1] = Q2'.Q1'.Q0' + Q2.Q'1.Q0$	

## 1.4. Waveform simulation

Test case: (-30, -40), (-5, 12), (8, 15), (12, -10)

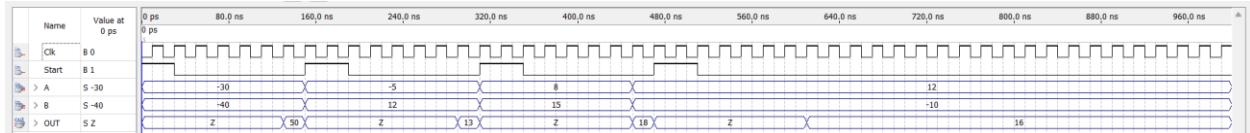


Figure 36: Task 1 Waveform

## TASK 2: SRA design with Graph-partitioning algorithm

### Graph-partitioning algorithm flow chart

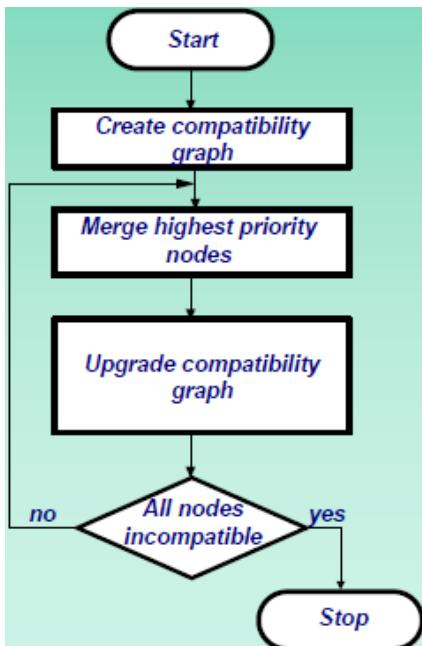


Figure 37: Graph Partitioning algorithm

### Graph-partitioning algorithm compatibility graph

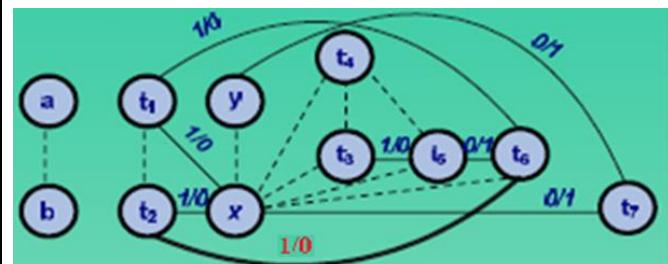


Figure 38: Initial compatibility graph

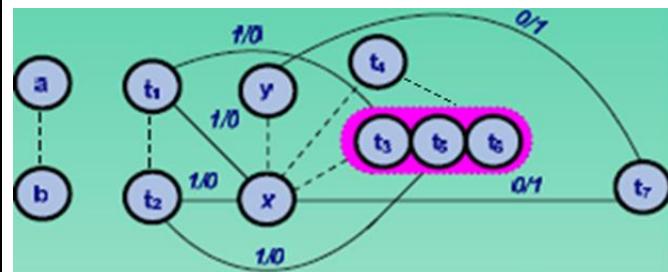


Figure 39: Compatibility graph after merging t3, t5 and t6

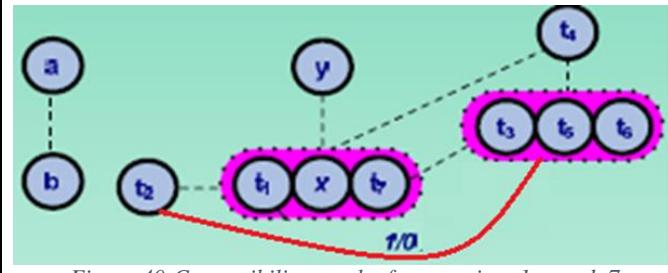


Figure 40: Compatibility graph after merging t1, x and t7

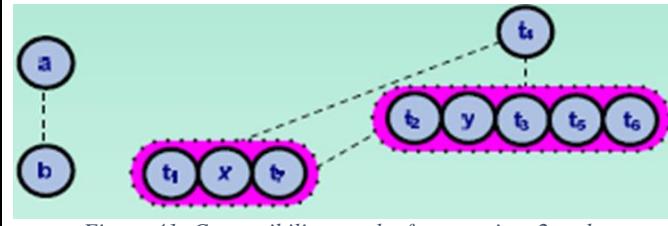


Figure 41: Compatibility graph after merging t2 and y

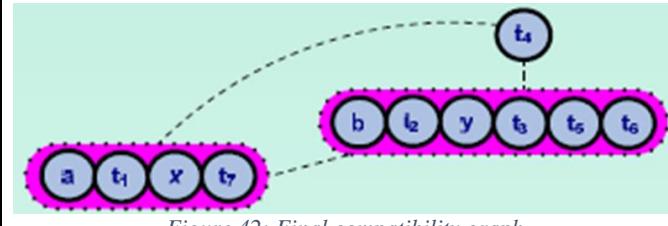
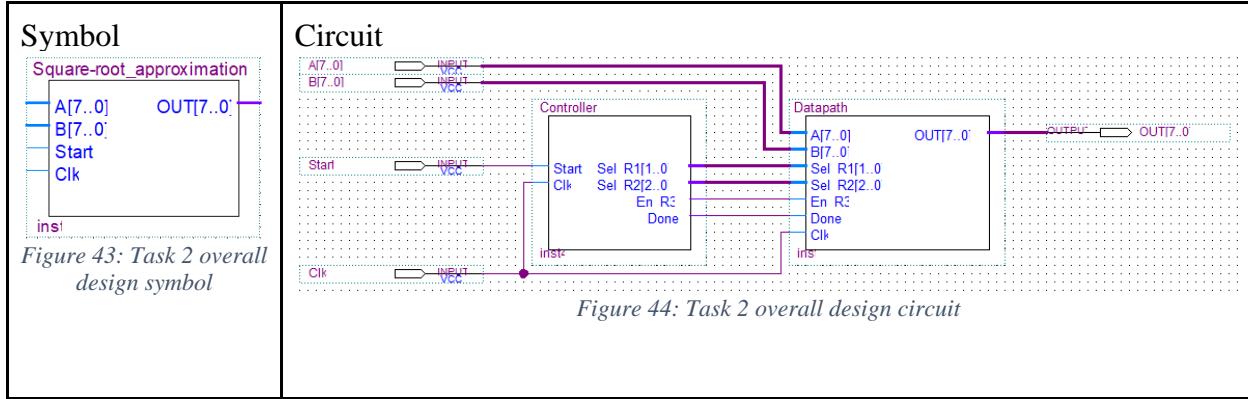


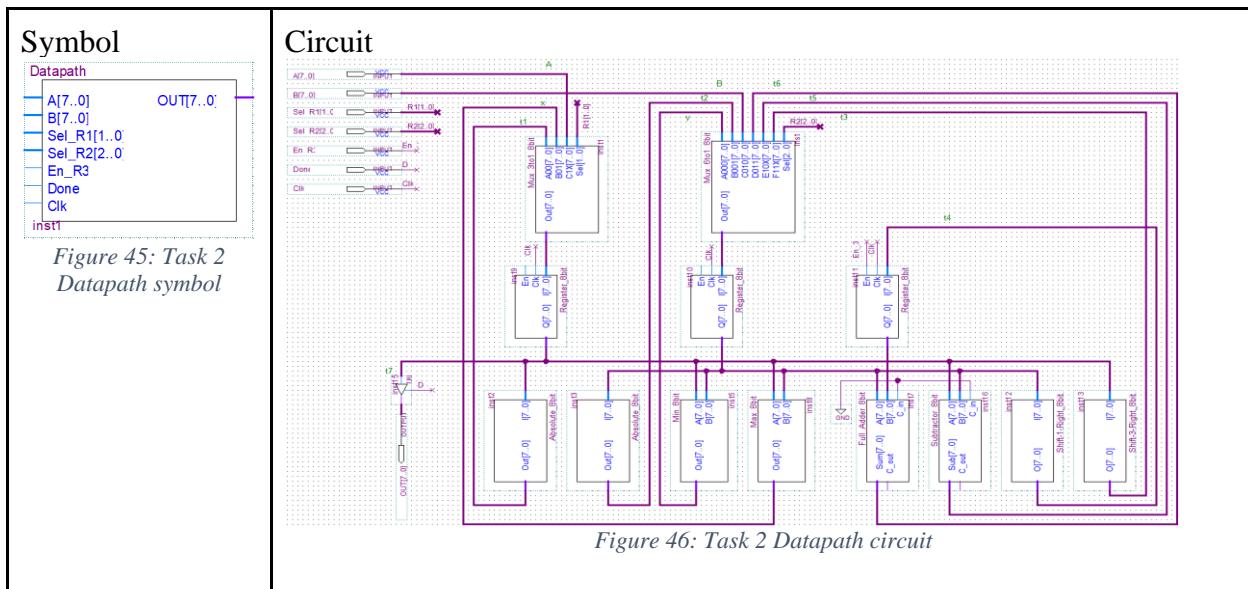
Figure 42: Final compatibility graph

- Compatibility graph: nodes and edges.
  - Node represents a variable.
  - Edge: incompatibility edge and priority edge.
    - Incompatibility edge (dashed line): connect 2 nodes indicates 2 variables with overlapping lifetimes.
    - Priority edge: connect 2 nodes indicates 2 non-overlapping lifetime variables that serve as the same source (input) or same destination (output) to the *same functional units or units in the used library*.
- Priority weight (s/d) on priority edge: the number of selector inputs can be saved
  - s is equal to the number of the *same functional units* that use both nodes as right operands (input).
  - d is equal to the number of the *same functional units* that generate results (output) for both nodes

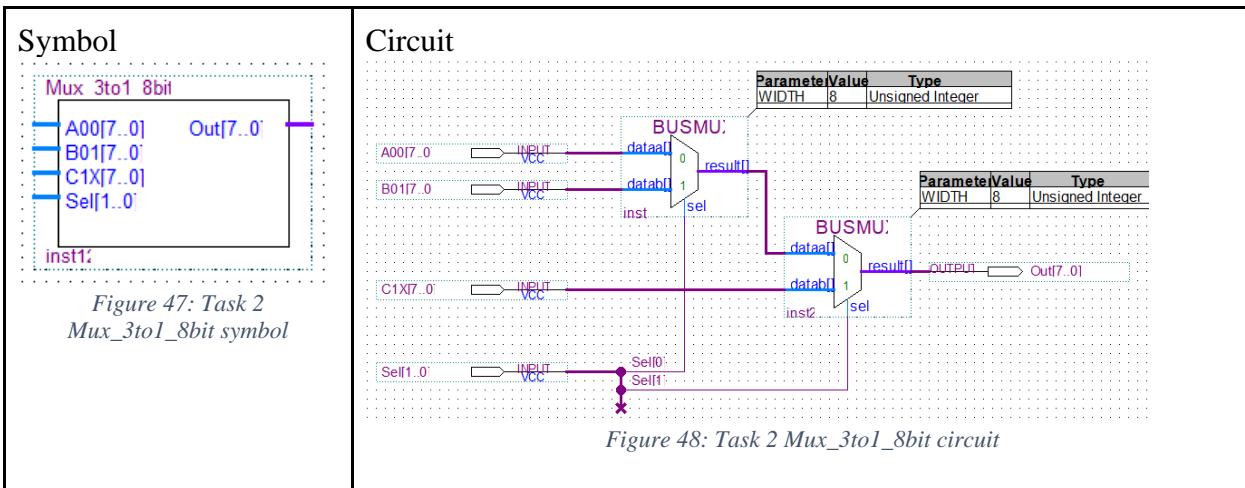
## 2.1. Overall Design



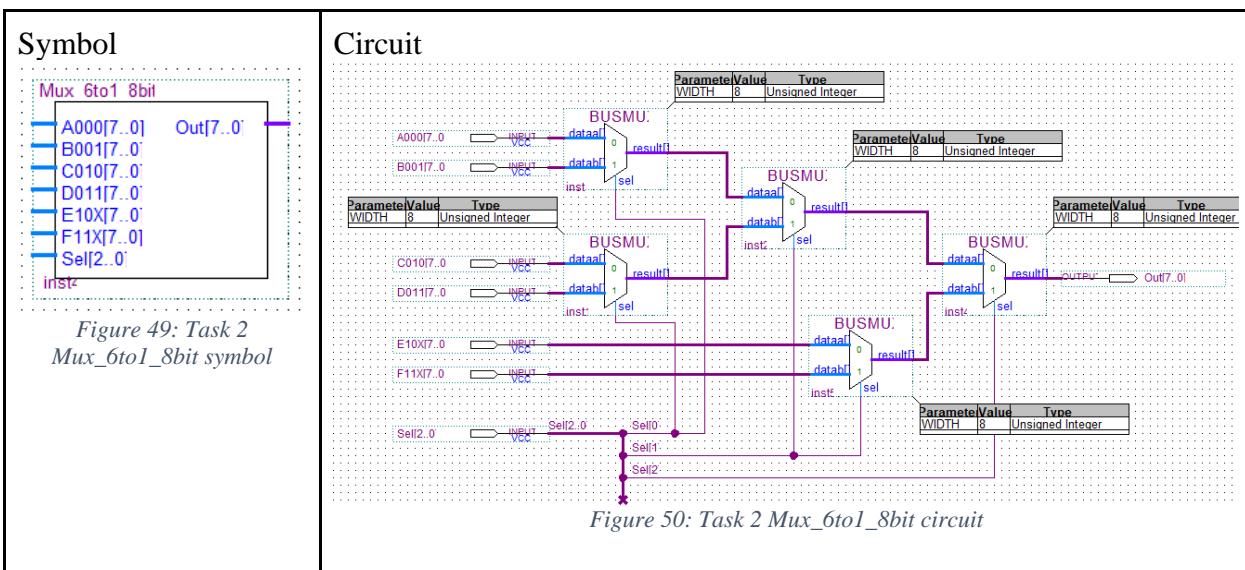
## 2.2. Datapath Design



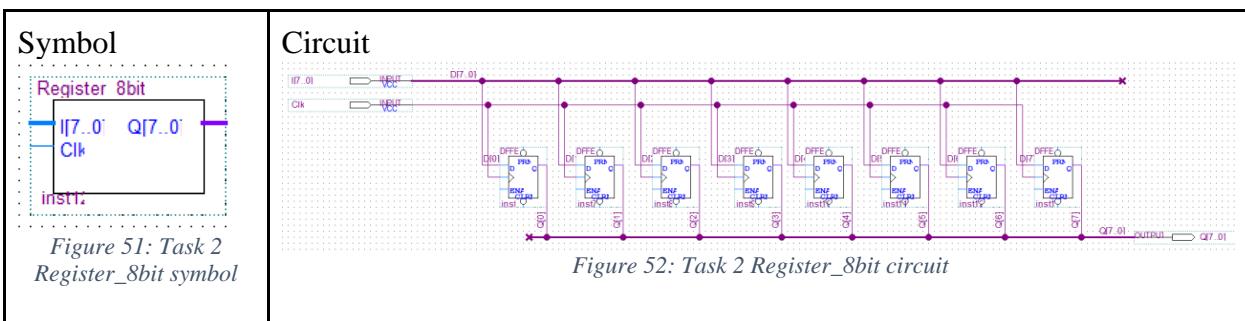
### 2.2.1. Mux\_3to1\_8bit



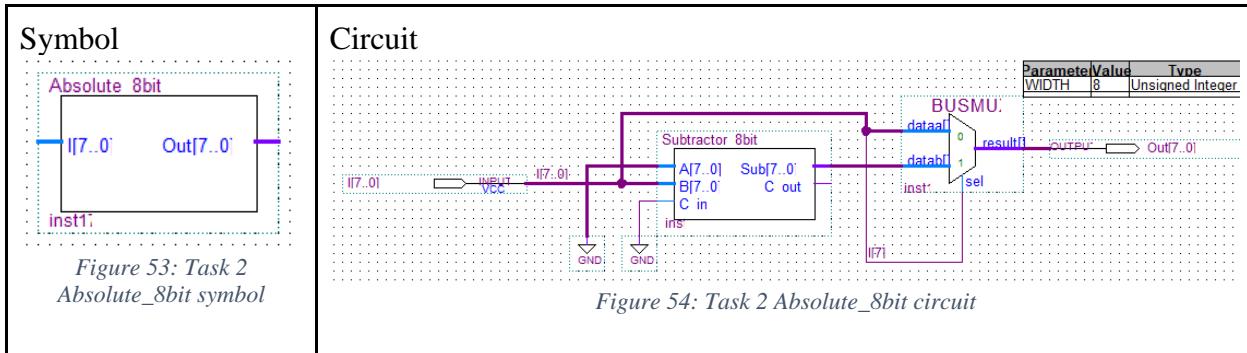
### 2.2.2. Mux\_6to1\_8bit



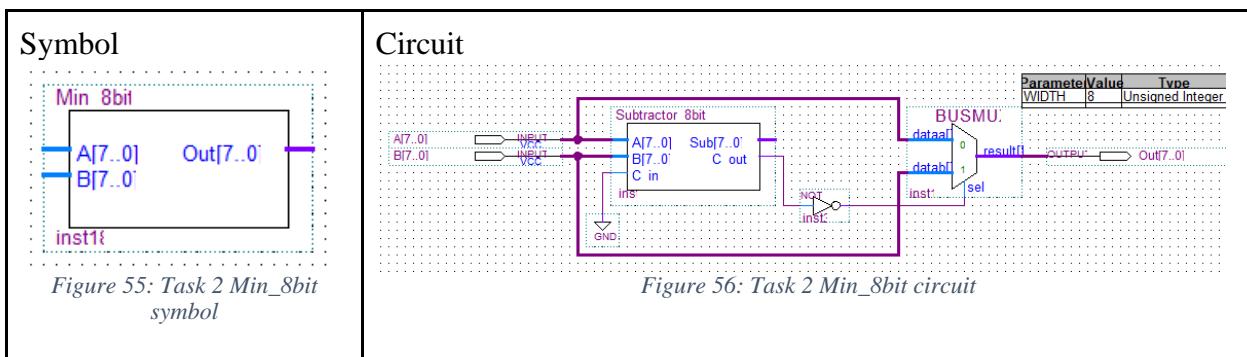
### 2.2.3. Register\_8bit



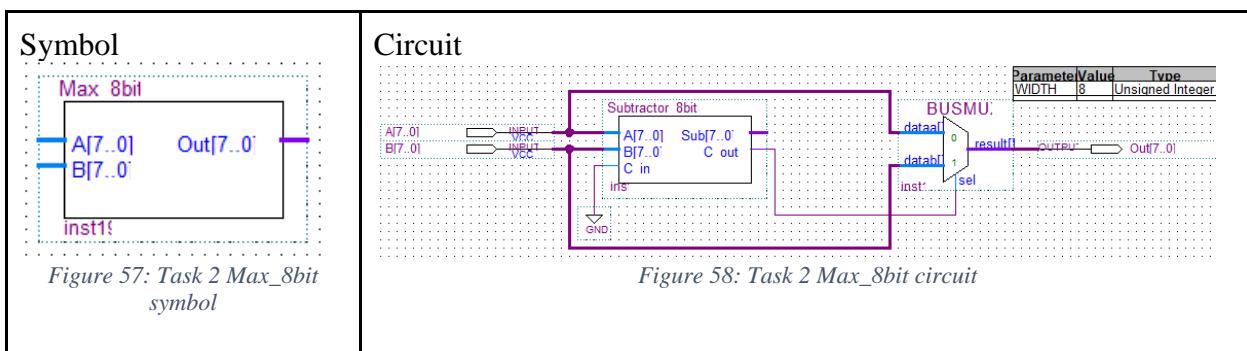
## 2.2.4. Absolute\_8bit



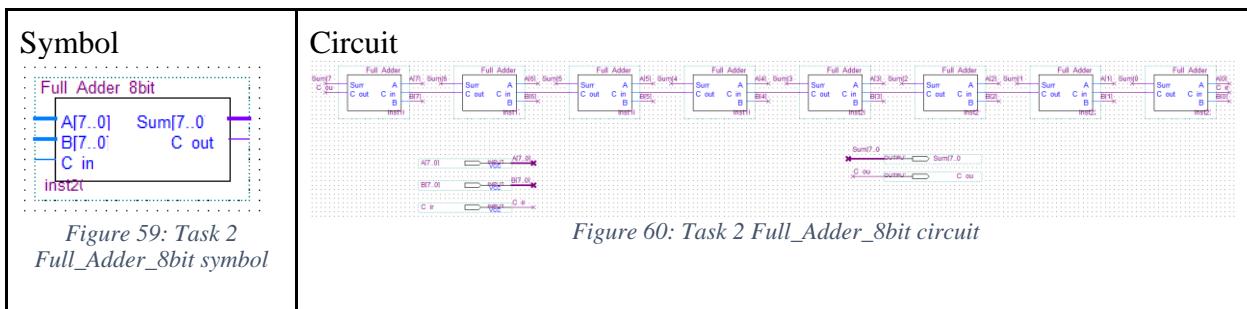
## 2.2.5. Min\_8bit



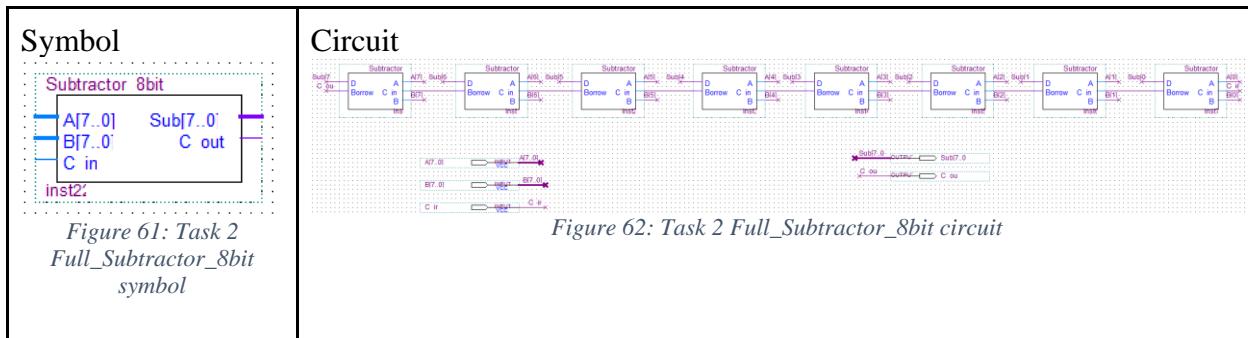
## 2.2.6. Max\_8bit



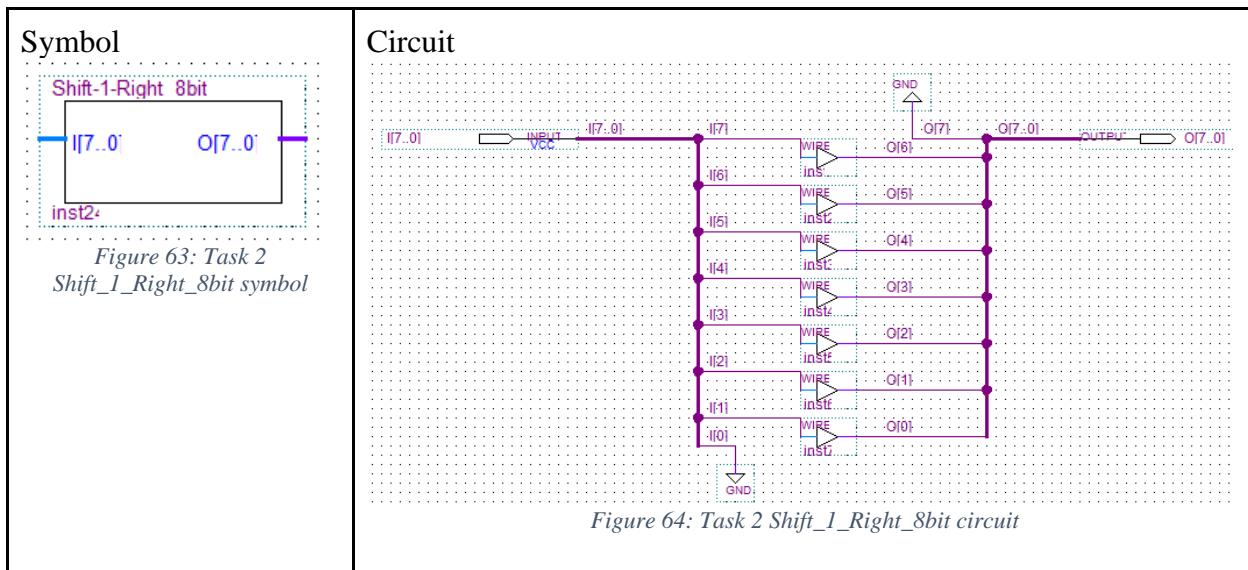
## 2.2.7. Full\_Adder\_8bit



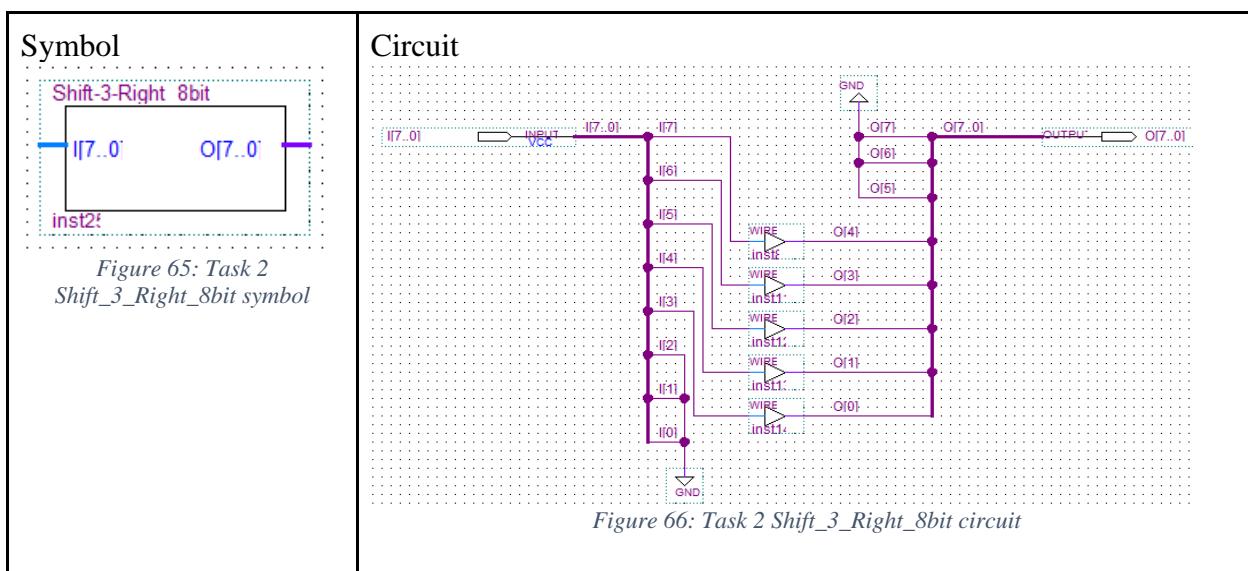
## 2.2.8. Subtractor\_8bit



## 2.2.9. Shift-1-Right\_8bit



## 2.2.10. Shift-3-Right\_8bit



## 2.3. Controller Design

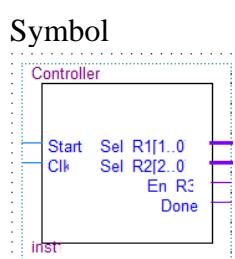


Figure 67: Task 2 Controller symbol

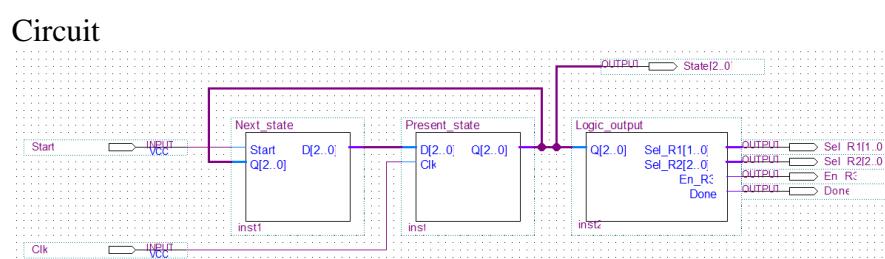


Figure 68: Task 2 Controller circuit

State Table

	Q2Q1Q0	Start/Reset	Q2+Q1+Q0+	Sel_R1	Sel_R2	En_R3	Done
s0	000	0	000	1X	010	0	0
	000	1	001	1X	010	0	0
s1	001	0	010	00	001	0	0
	001	1	010	00	001	0	0
s2	010	0	011	01	000	0	0
	010	1	011	01	000	0	0
s3	011	0	100	01	11X	1	0
	011	1	100	01	11X	1	0
s4	100	0	101	01	10X	0	0
	100	1	101	01	10X	0	0
s5	101	0	110	01	011	0	0
	101	1	110	01	011	0	0
s6	110	0	111	01	011	0	0
	110	1	111	01	011	0	0
s7	111	0	000	01	011	0	1
	111	1	000	01	011	0	1

State Diagram

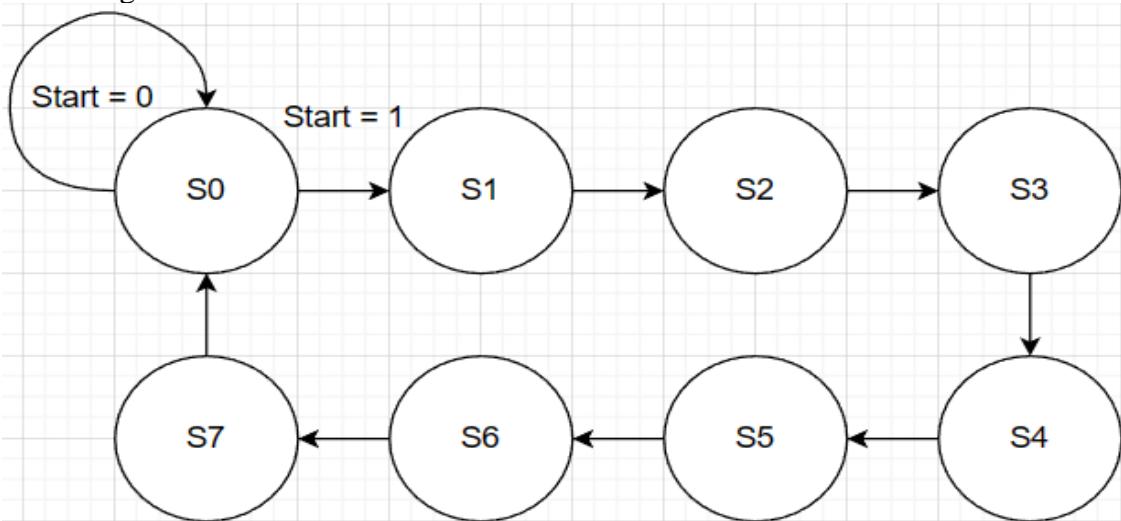
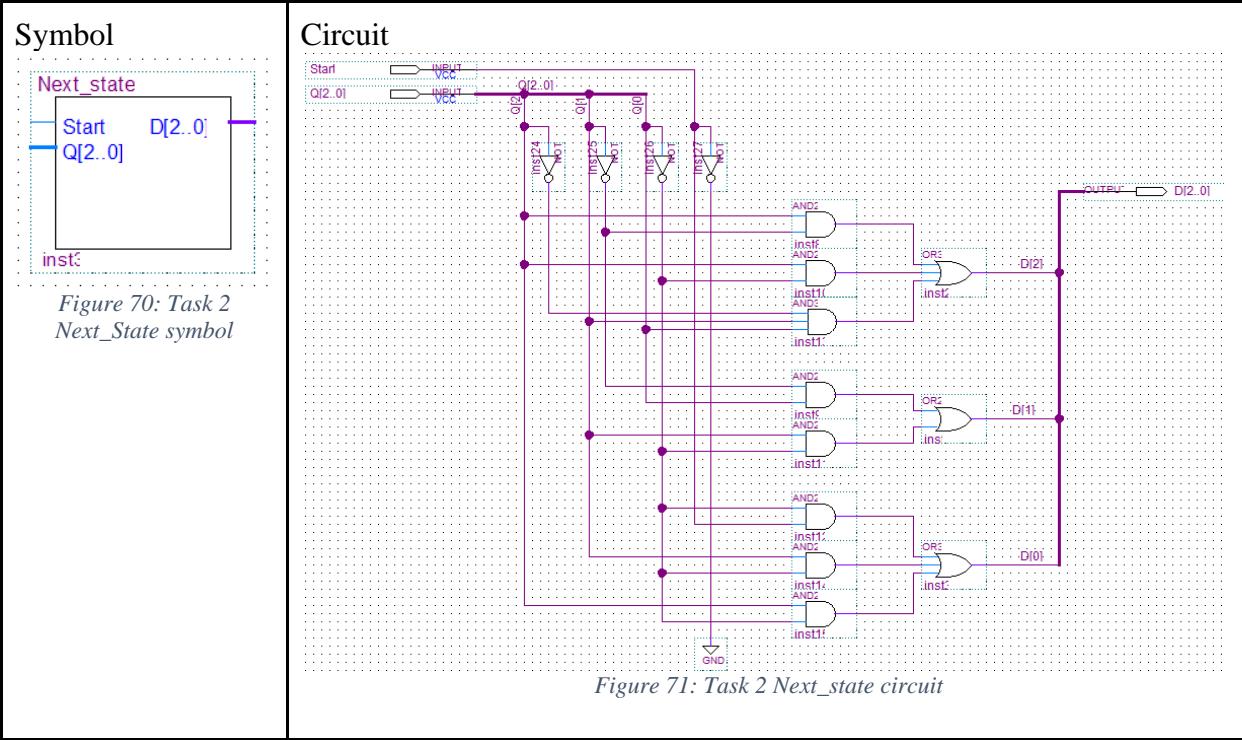


Figure 69: Task 2 state diagram

### 2.3.1. Next\_state



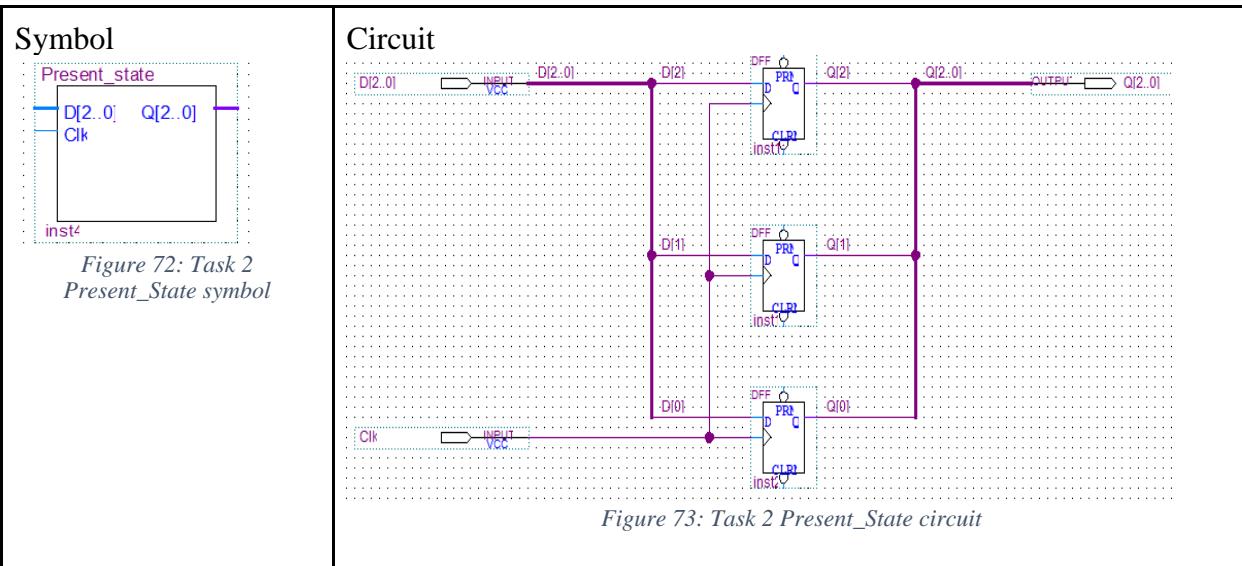
**Next\_state equation**

$$Q2+ = Q2.Q1' + Q2.Q0' + Q2'.Q1.Q0$$

$$Q1+ = Q1'.Q0 + Q1.Q0'$$

$$Q0+ = Q0'.Start + Q1.Q0' + Q2.Q0'$$

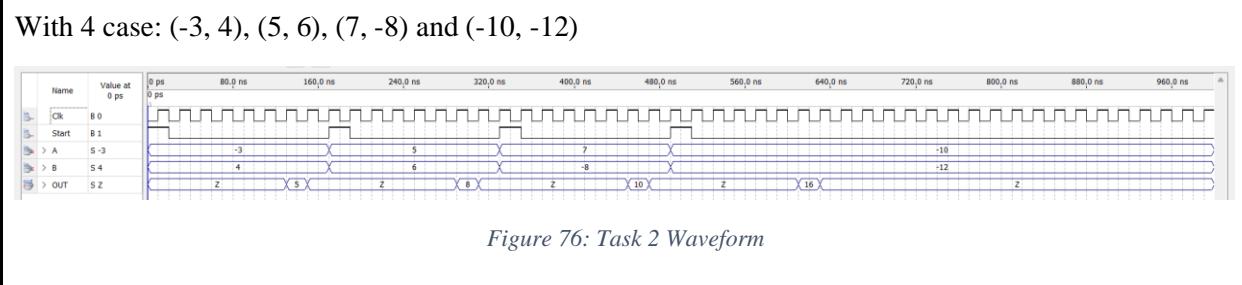
### 2.3.2. Present\_state



### 2.3.3. Logic\_output

<p><b>Symbol</b></p> <p>Figure 74: Task 2 Logic_Output symbol</p>	<p><b>Circuit</b></p> <p>Figure 75: Task 2 Logic_output circuit</p>
<p><b>Logic_output equation</b></p> $\begin{aligned} \text{Sel\_R1}[1] &= Q2'.Q1'.Q0' \\ \text{Sel\_R1}[0] &= Q2 + Q1 \\ \text{Sel\_R2}[2] &= Q1.Q0 + Q2.Q1'.Q0' \\ \text{Sel\_R2}[1] &= Q2'.Q1'.Q0' + Q2.Q1 + Q2.Q0 + Q1.Q0 \end{aligned}$ $\begin{aligned} \text{Sel\_R2}[0] &= Q2 + Q0 \\ \text{En\_R3} &= Q1.Q0 \\ \text{Done} &= Q2.Q1.Q0 \end{aligned}$	

### 2.4. Waveform simulation

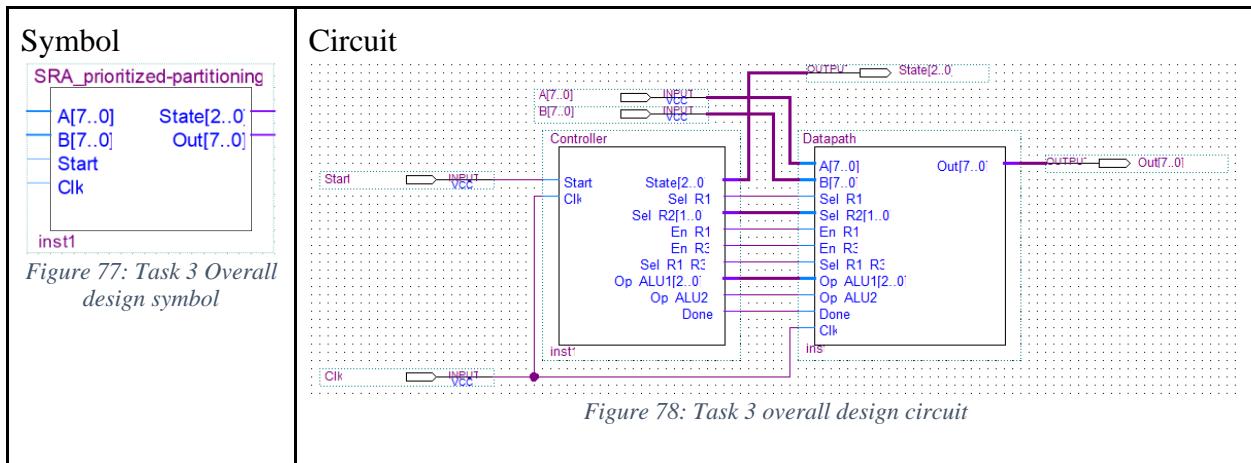


## TASK 3: SRA design with prioritized partitioning

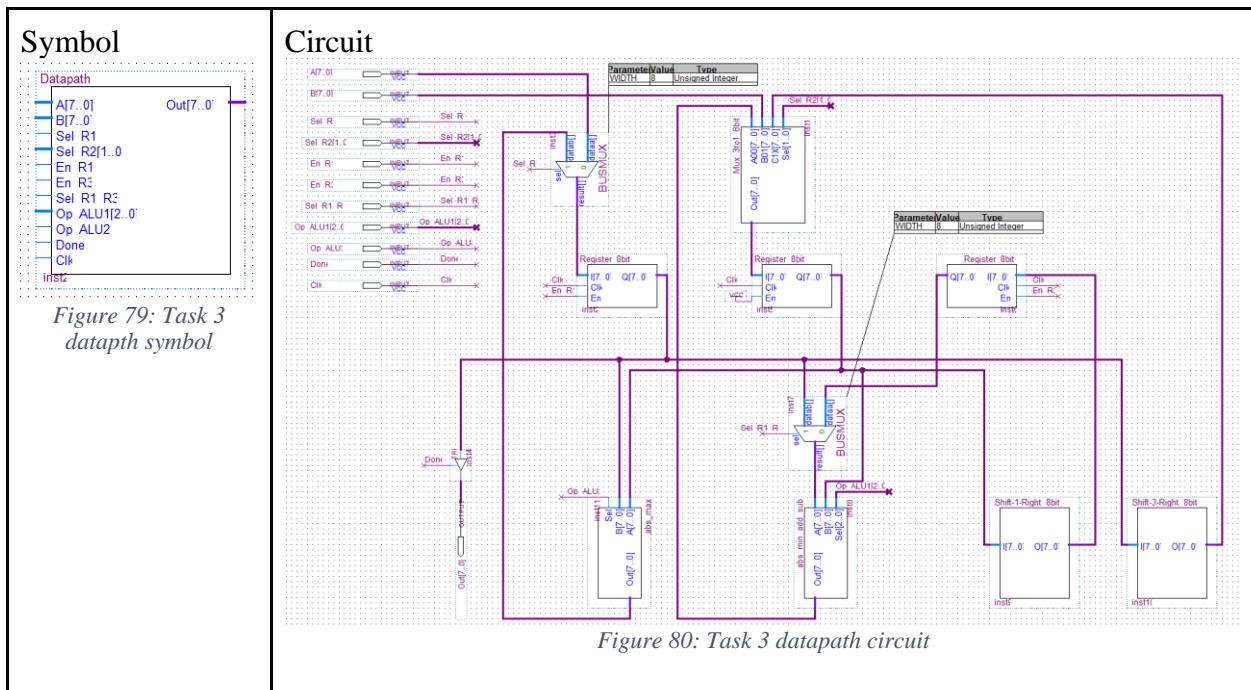
### Basic theory:

- Group non-concurrent operations
- Each group shares one functional unit
- Sharing reduces number of functional units
- Prioritized grouping by reducing connectivity
- Clustering algorithm used for grouping

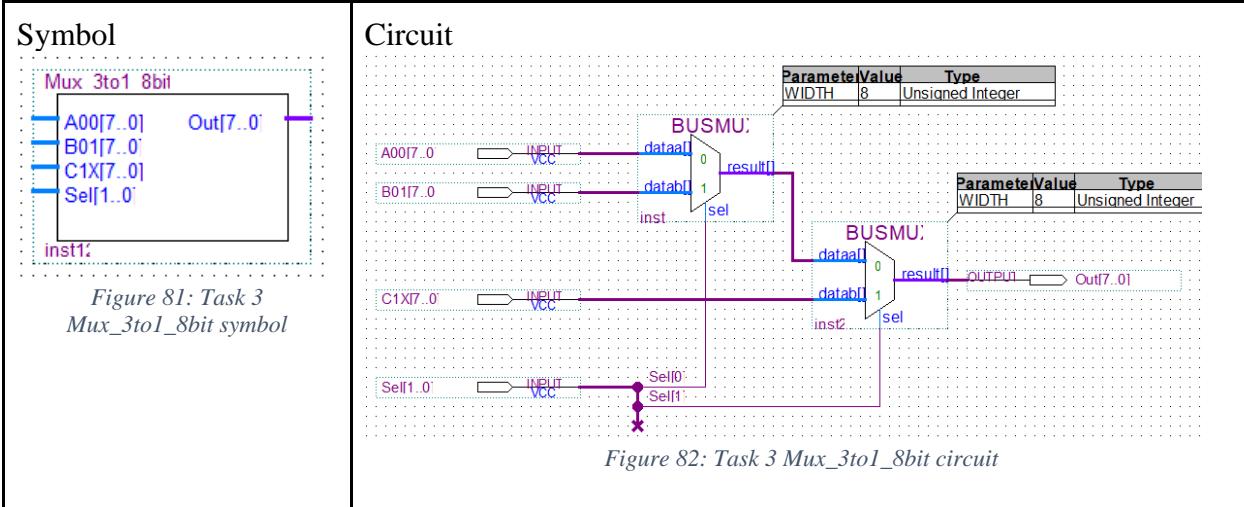
### 3.1. Overall Design



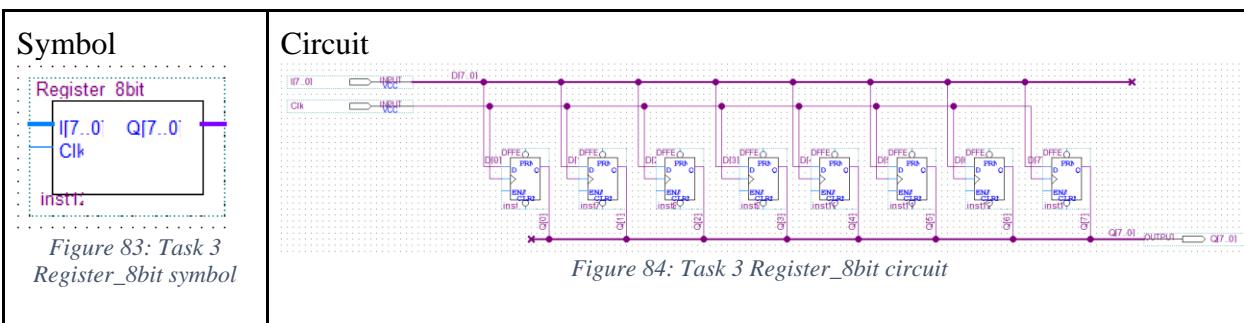
### 3.2. Datapath Design



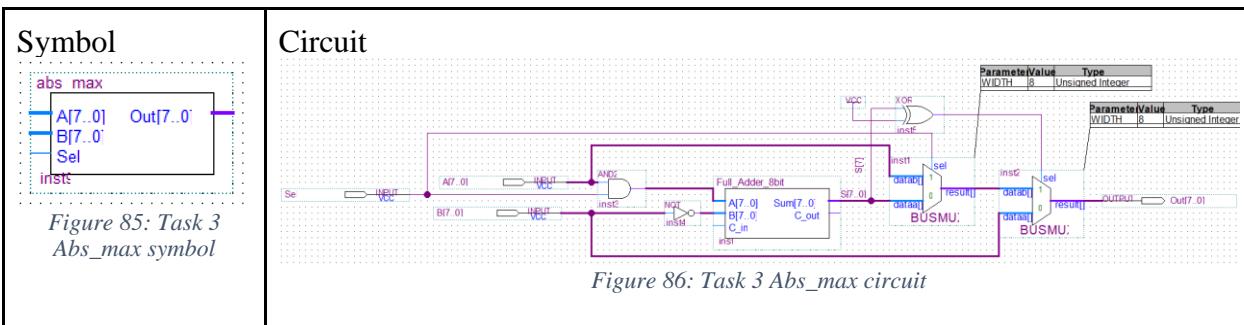
### 3.2.1. Mux\_3to1\_8bit



### 3.2.2. Register\_8bit



### 3.2.3. abs\_max



Operation table

Sel	Function
0	Abs(B)
1	Max

### 3.2.4. abs\_min\_add\_sub

Symbol	Circuit
<p>Figure 87: Task 3 Abs_Min_Add_Sub symbol</p> <pre> abs_min_add_sub   A[7..0]  Out[7..0]   B[7..0]   Sel[2..0] inst1 </pre>	<p>Figure 88: Task 3 Abs_Min_Add_Sub circuit</p>

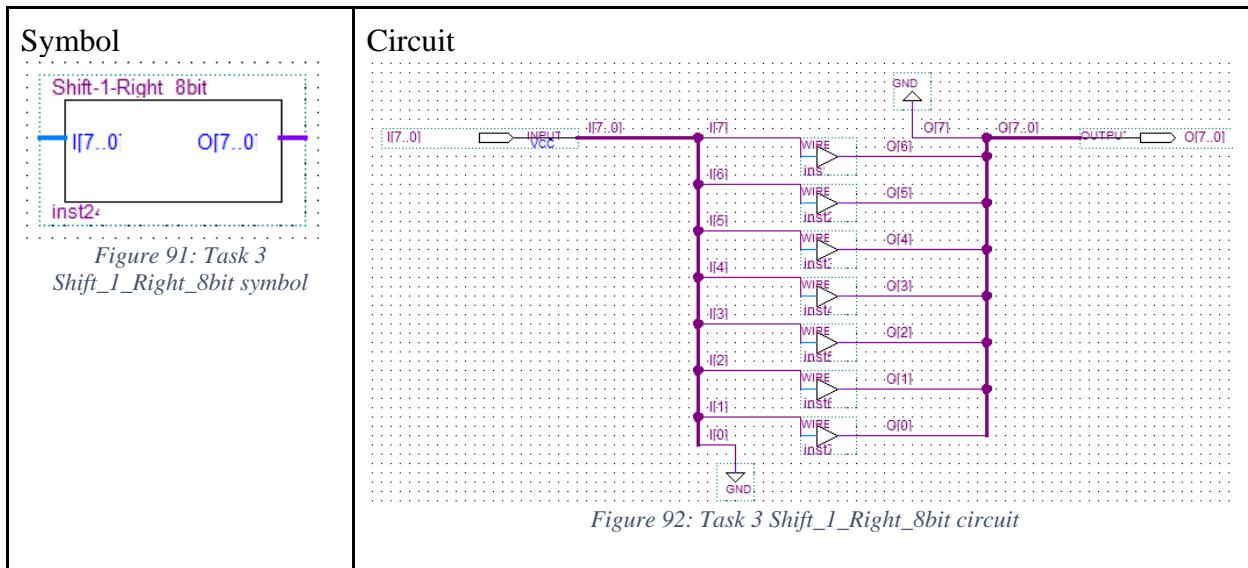
  

Operation table			
Sel1	Sel0	Signal bit	Function
0	0	0	Abs
0	0	1	
0	1	0	Min
0	1	1	
1	0	0	Add
1	0	1	
1	1	0	Sub
1	1	1	

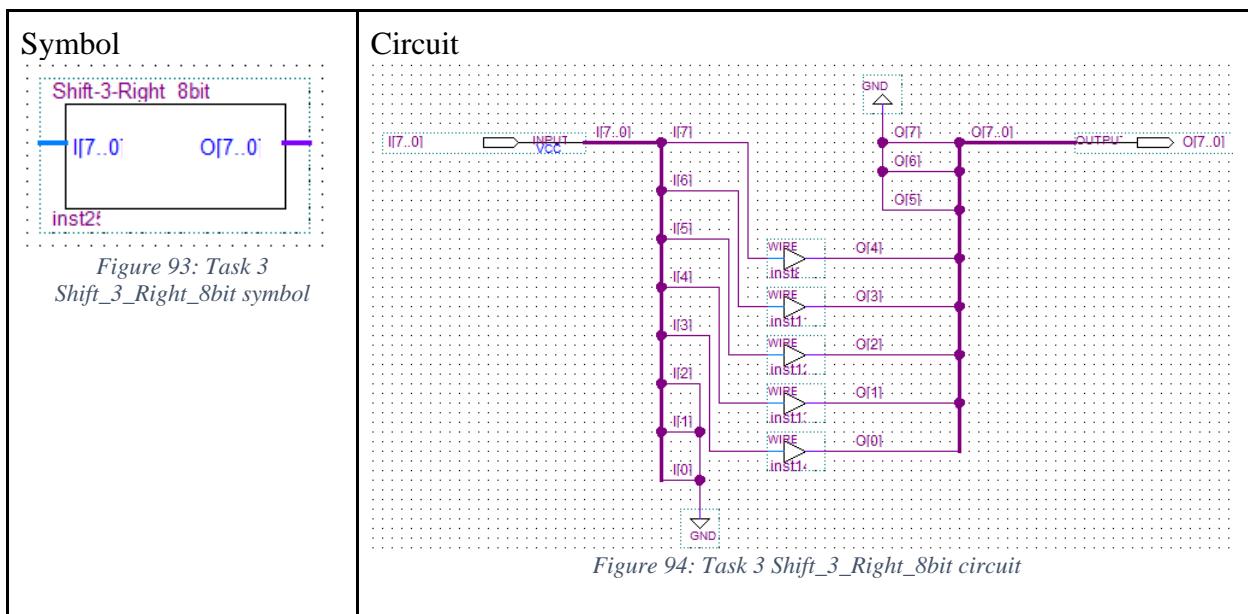
### 3.2.5. Full\_Adder\_8bit

Symbol	Circuit
<p>Figure 89: Task 3 Full_Adder_8bit symbol</p> <pre> Full Adder 8bit   A[7..0]  Sum[7..0]   B[7..0]   C in inst2 </pre>	<p>Figure 90: Task 3 Full_Adder_8bit circuit</p>

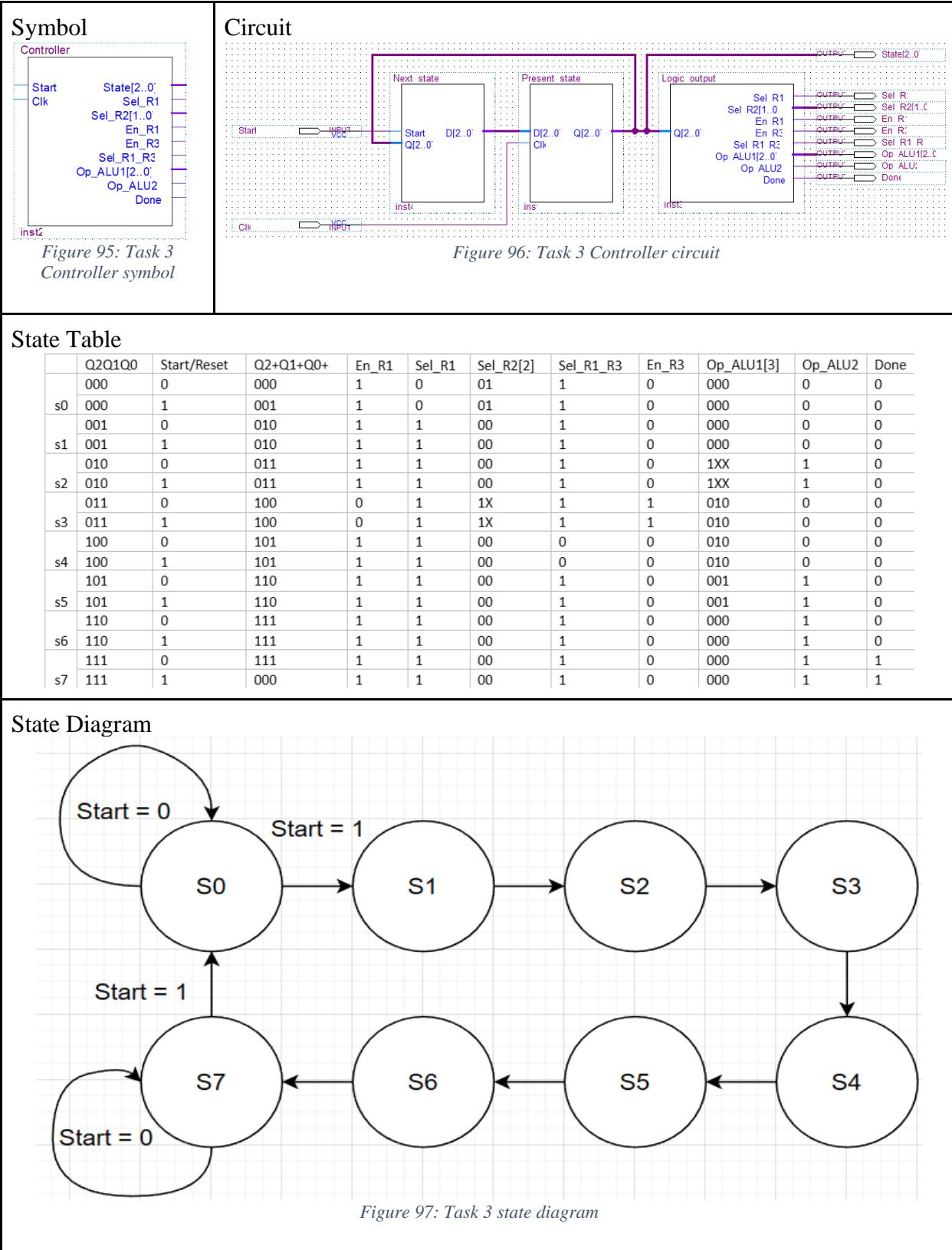
### 3.2.6. Shift-1-Right\_8bit



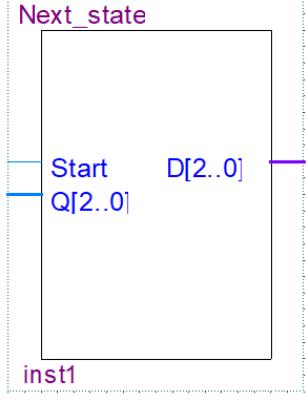
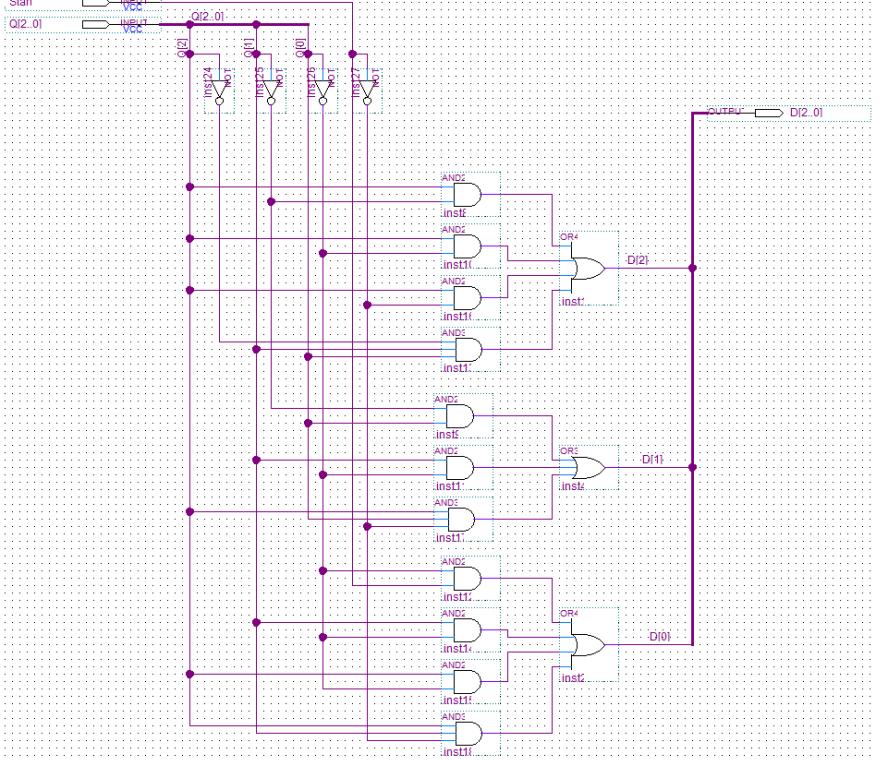
### 3.2.7. Shift-3-Right\_8bit



### 3.3. Controller Design

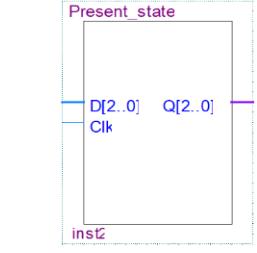
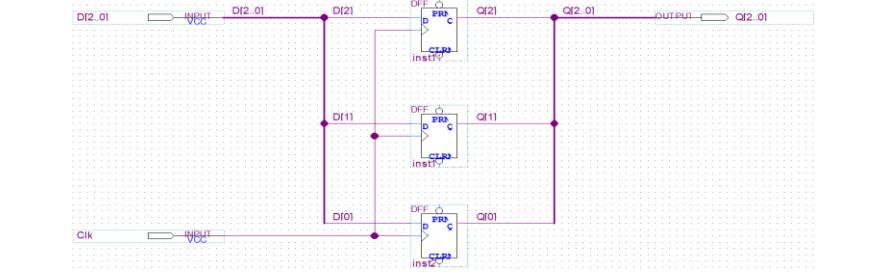


### 3.3.1. Next\_state

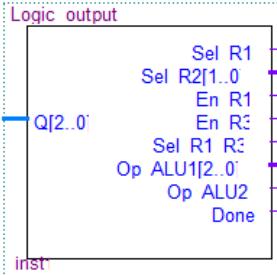
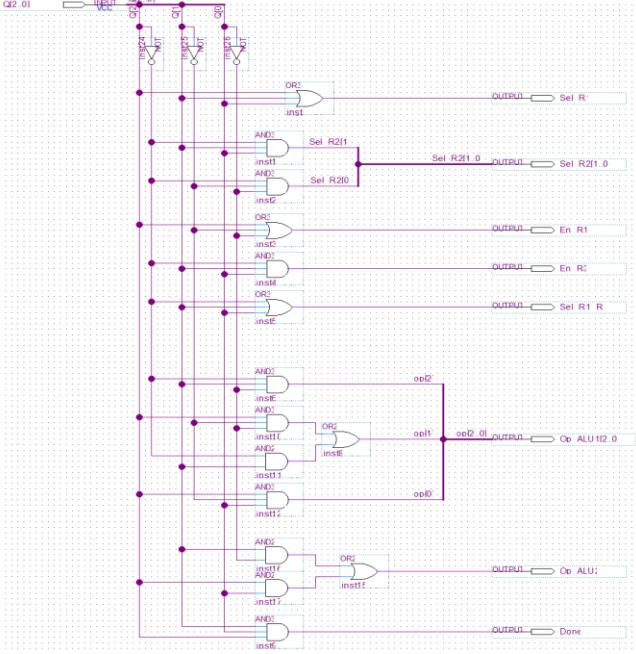
<p>Symbol</p>  <p>Figure 98: Task 3 Next_State symbol</p>	<p>Circuit</p>  <p>Figure 99: Task 3 Next_State circuit</p>
--	--

Next\_state equation  
 $Q2+ = Q2.Q1' + Q2.Q0' + Q2.Start' + Q2'.Q1.Q0$   
 $Q1+ = Q1'.Q0 + Q1.Q0' + Q2.Q0.Start'$   
 $Q0+ = Q0'.Start + Q1.Q0' + Q2.Q0' + Q2.Q1.Start'$

### 3.3.2. Present\_state

<p>Symbol</p>  <p>Figure 100: Task 3 Present_State symbol</p>	<p>Circuit</p>  <p>Figure 101: Task 3 Present_State circuit</p>
--	---

### 3.3.3. Logic\_output

Symbol	Circuit
 <p>Figure 102: Task 3 Logic_Output symbol</p>	 <p>Figure 103: Task 3 Logic_Output circuit</p>

Logic_output equation	
$En\_R1 = Q2 + Q1' + Q0'$	$Op\_ALU1[2] = Q2'.Q1.Q0'$
$En\_R3 = Q2'.Q1.Q0$	$Op\_ALU1[1] = Q2.Q1'.Q0' + Q2'.Q1$
$Sel\_R1 = Q2 + Q1 + Q0$	$Op\_ALU1[0] = Q2.Q1'.Q0$
$Sel\_R2[1] = Q2'.Q1.Q0$	$Op\_ALU2 = Q1.Q0' + Q2.Q0$
$Sel\_R2[0] = Q2'.Q1'.Q0'$	$Done = Q2.Q1.Q0$
$Sel\_R1\_R3 = Q2' + Q1 + Q0$	

### 3.4. Waveform simulation

With 4 case: (-3, 4), (-6, 8), (5, 12) and (15, -8)

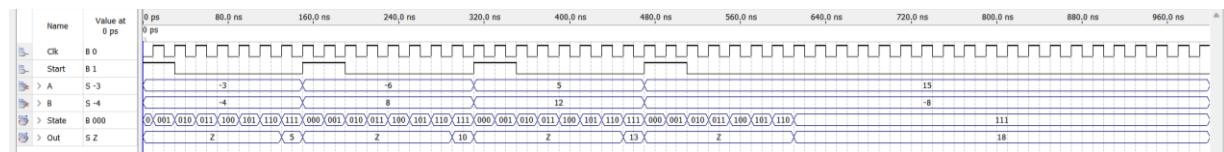


Figure 104: Task 3 Waveform

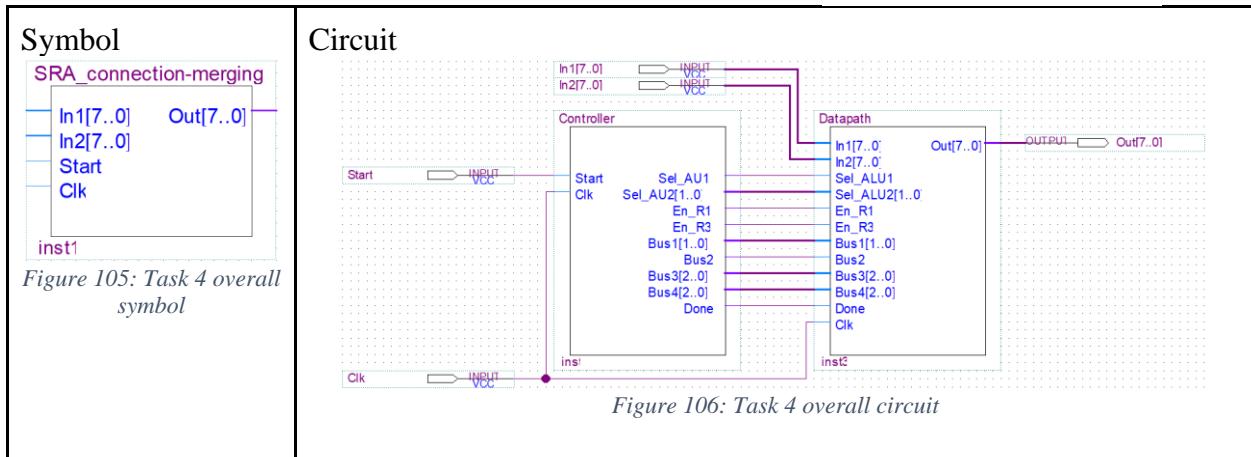
## TASK 4: SRA design with connection merging

**Connection sources:** outputs of register and functional units / **input** of circuit

**Connection destinations:** inputs of register and functional units

- **Edge:** incompatibility edge and priority edge
  - + **Incompatibility edge** (dashed line): connect 2 nodes whenever their corresponding connections *do not originate from* the same source, but are used at the same time.
  - + **Priority edge**: connect 2 nodes whenever their corresponding connections have a common source or a common destination.

### 4.1. Overall Design



### 4.2. Datapath Design



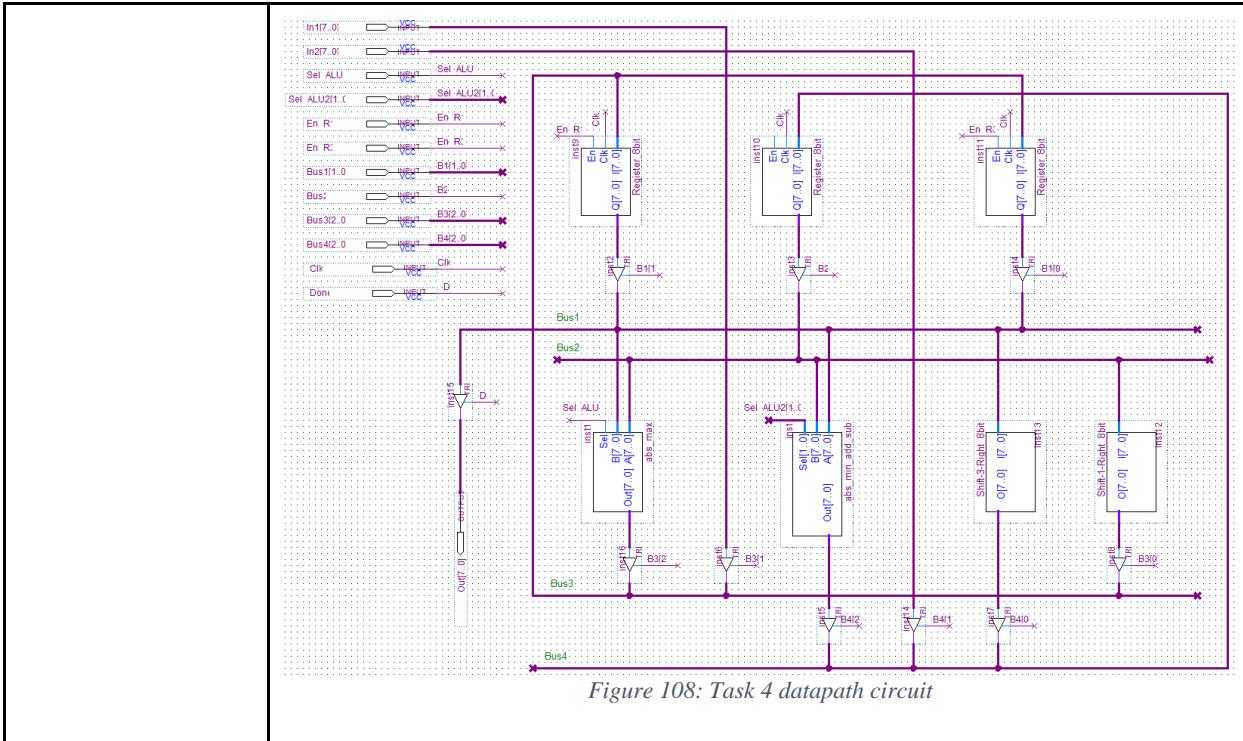


Figure 108: Task 4 datapath circuit

#### 4.2.1. Register\_8bit

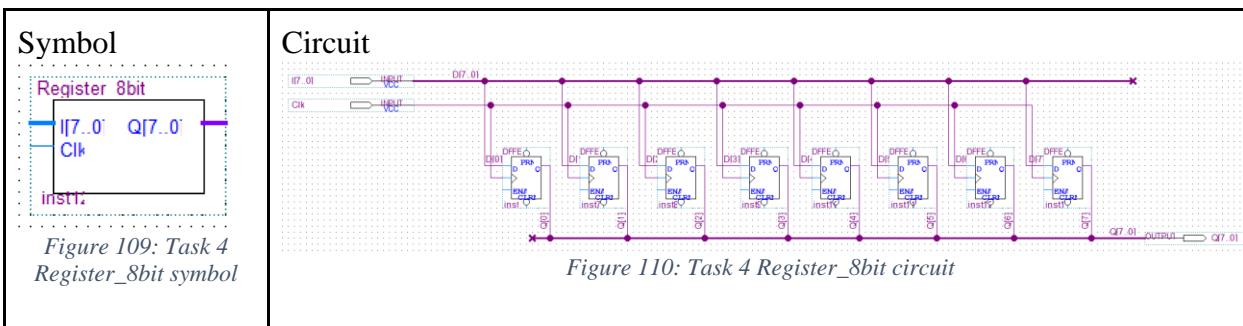


Figure 109: Task 4 Register\_8bit symbol

Figure 110: Task 4 Register\_8bit circuit

#### 4.2.2. abs\_max

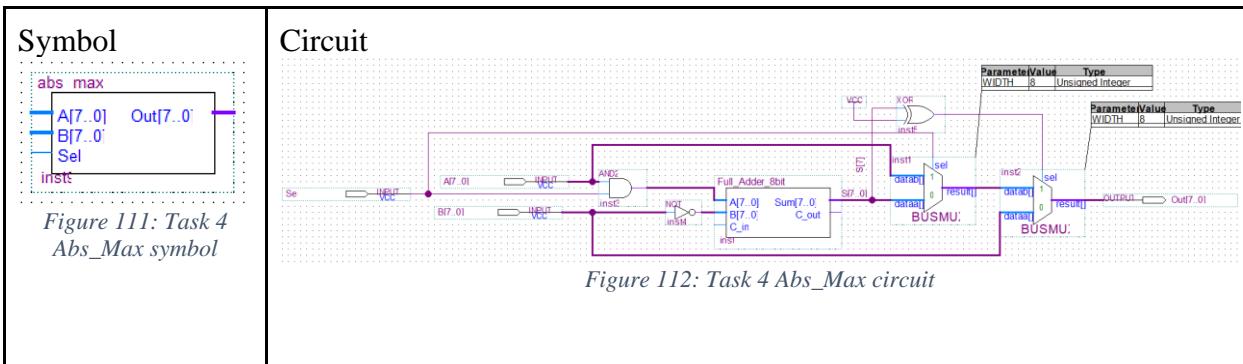


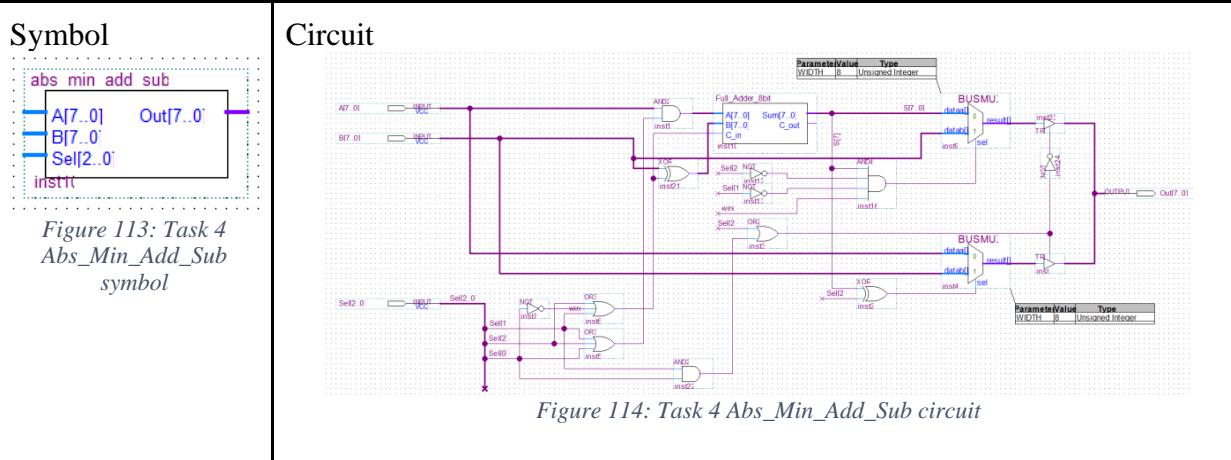
Figure 111: Task 4 Abs\_Max symbol

Figure 112: Task 4 Abs\_Max circuit

Operation table

Sel	Function
0	abs(B)
1	max

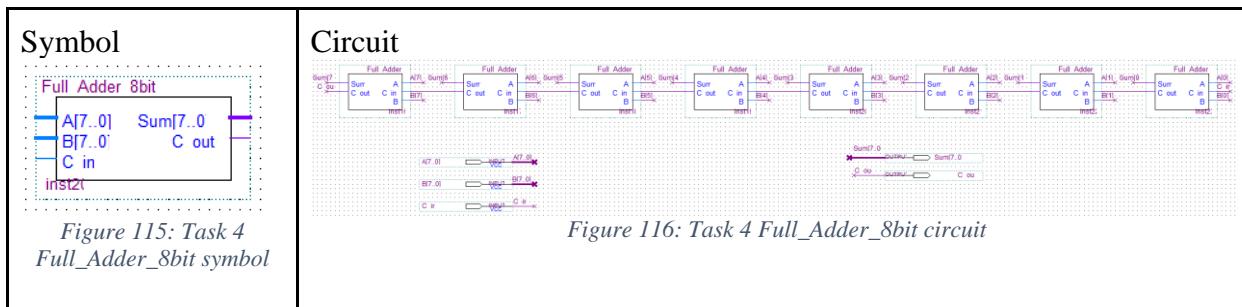
#### 4.2.3. abs\_min\_add\_sub



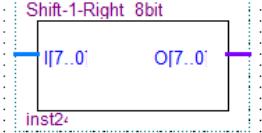
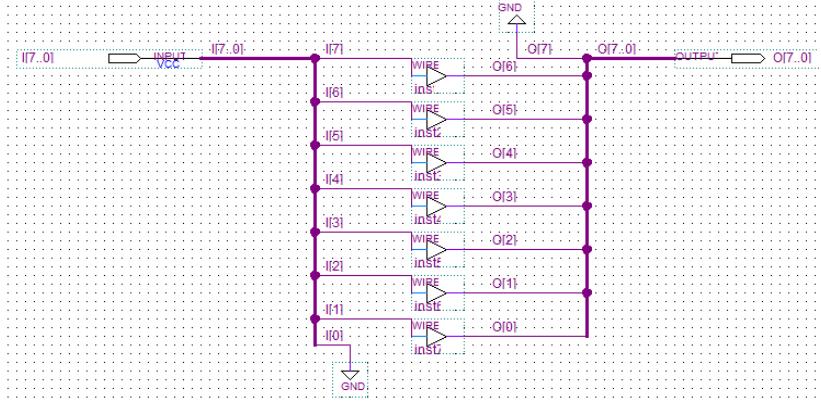
Operation table

Sel	Function
00	abs
01	min
10	add
11	sub

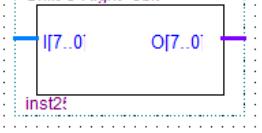
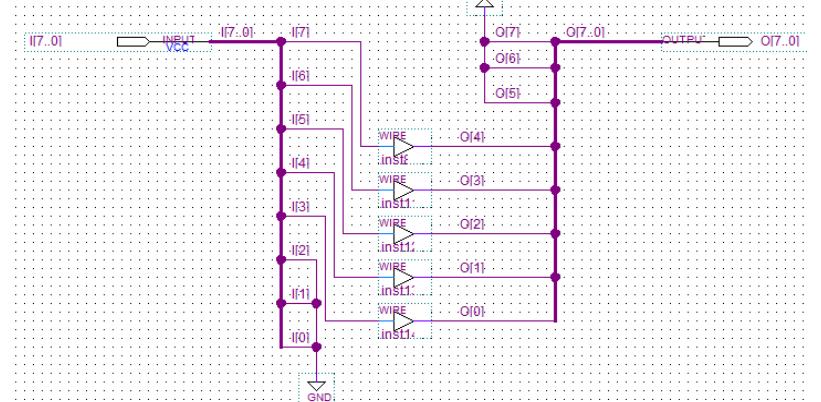
#### 4.2.4. Full\_Adder\_8bit



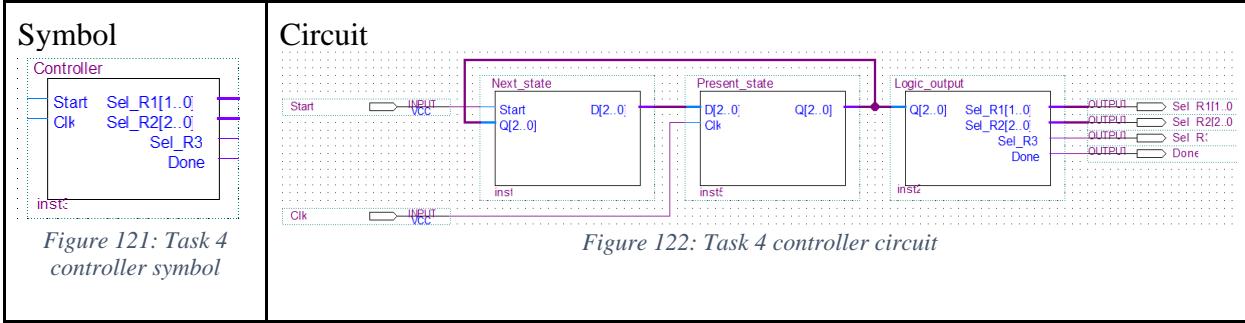
#### 4.2.5. Shift-1-Right\_8bit

Symbol	Circuit
 <p>Figure 117: Task 4 Shift_1_Right_8bit symbol</p>	 <p>Figure 118: Task 4 Shift_1_Right_8bit circuit</p>

#### 4.2.6. Shift-3-Right\_8bit

Symbol	Circuit
 <p>Figure 119: Task 4 Shift_3_Right_8bit symbol</p>	 <p>Figure 120: Task 4 Shift_3_Right_8bit circuit</p>

### 4.3. Controller Design



State Table

	Q2Q1Q0	Start	Q2+Q1+Q0+	Sel_ALU1	Sel_ALU2[2]	En_R1	En_R3	Bus1[2]	Bus2	Bus3[3]	Bus4[3]
s0	000	0	000	0	00	1	0	00	0	010	010
	000	1	001	0	00	1	0	00	0	010	010
	001	0	010	0	00	1	0	10	1	100	100
s1	001	1	010	0	00	1	0	10	1	100	100
	010	0	011	1	01	1	0	10	1	100	100
s2	010	1	011	1	01	1	0	10	1	100	100
	011	0	100	0	00	0	1	10	1	001	001
s3	011	1	100	0	00	0	1	10	1	001	001
	100	0	101	0	11	0	0	10	1	000	100
s4	100	1	101	0	11	0	0	10	1	000	100
	101	0	110	0	10	0	0	01	1	000	100
s5	101	1	110	0	10	0	0	01	1	000	100
	110	0	111	1	00	1	0	10	1	100	000
s6	110	1	111	1	00	1	0	10	1	100	000
	111	0	000	0	00	0	0	10	1	100	000
s7	111	1	000	0	00	0	0	10	1	100	000

State Diagram

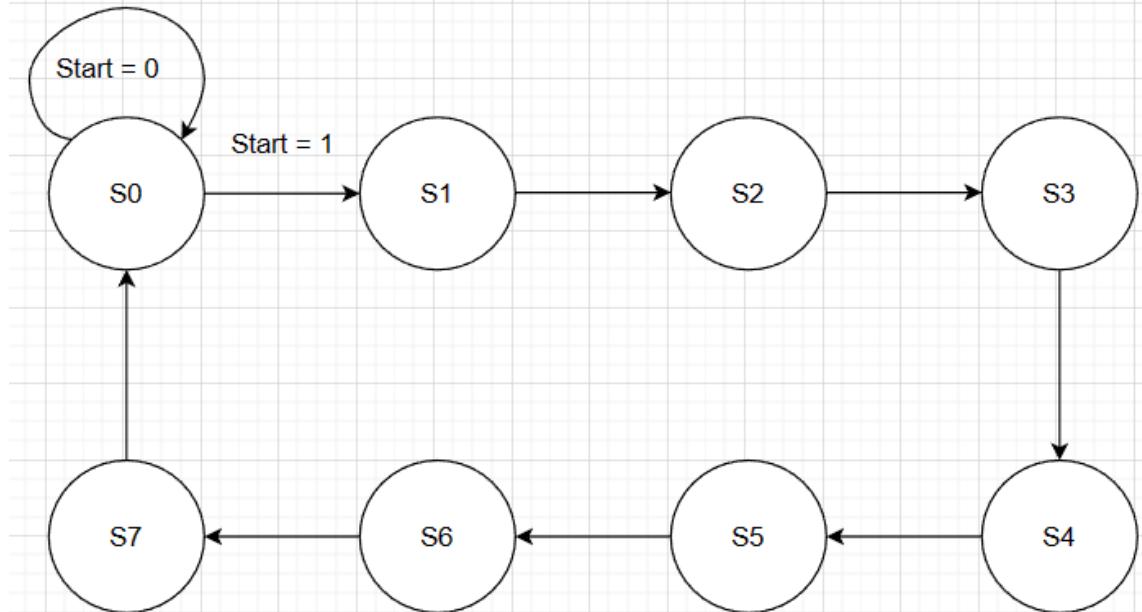
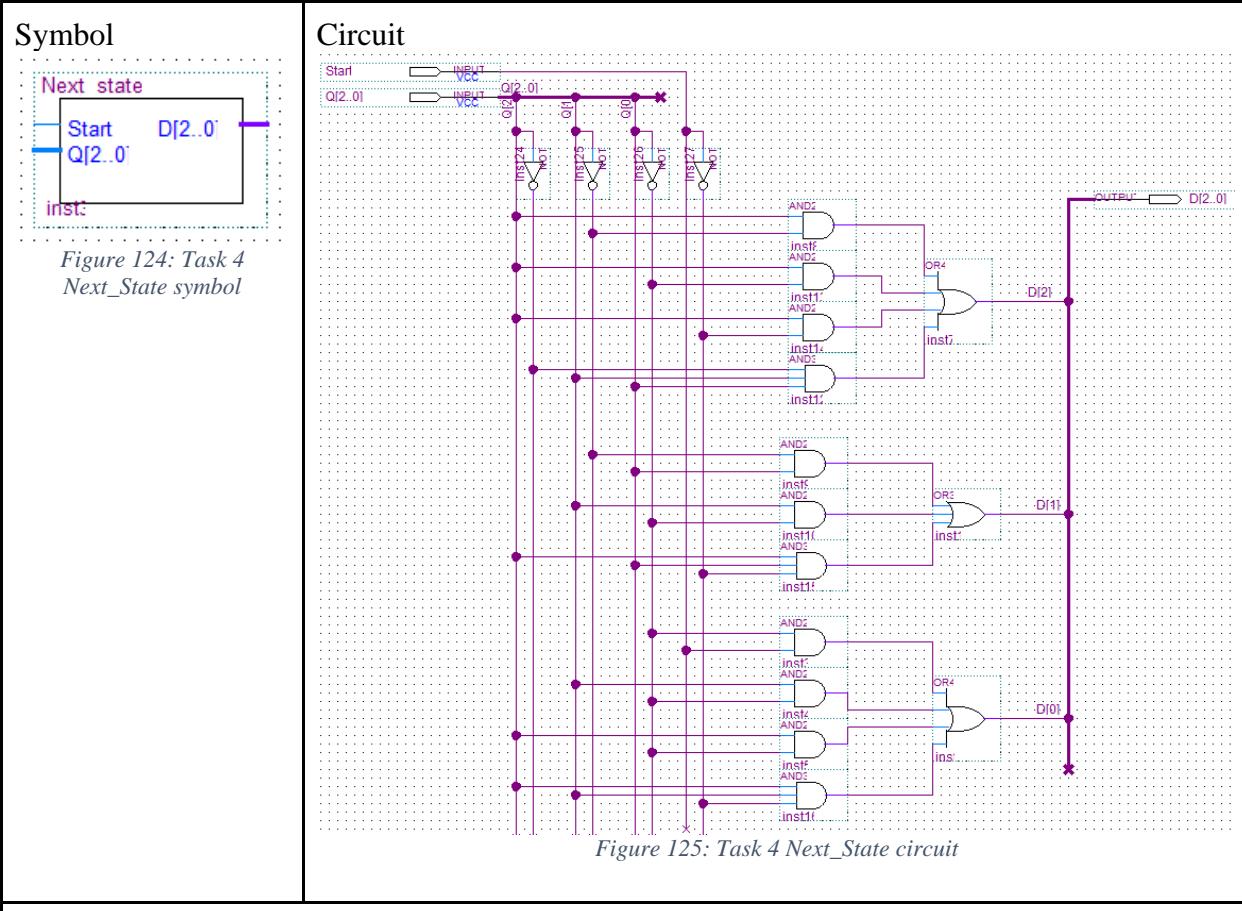


Figure 123: Task 4 state diagram

### 4.3.1. Next\_state



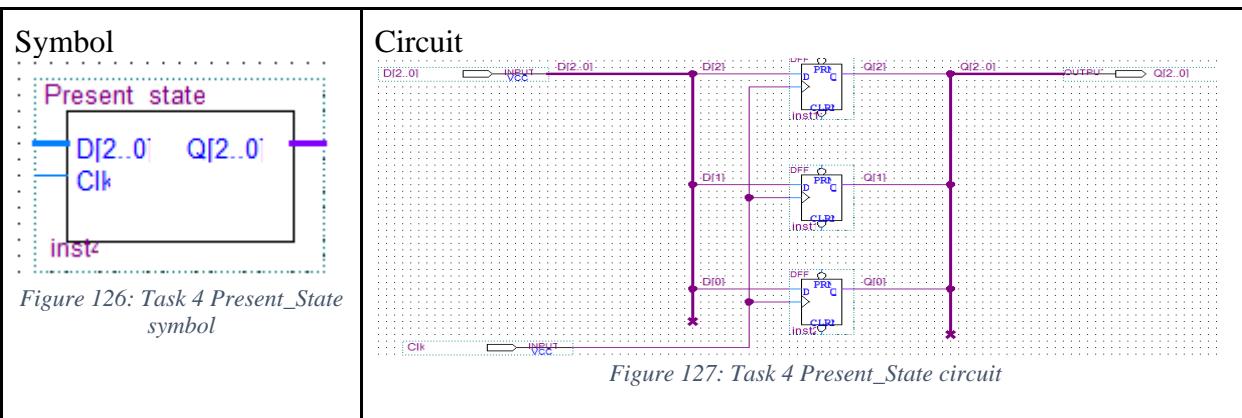
Next\_state equation

$$Q2+ = Q2.Q1' + Q2.Q0' + Q2'.Q1.Q0$$

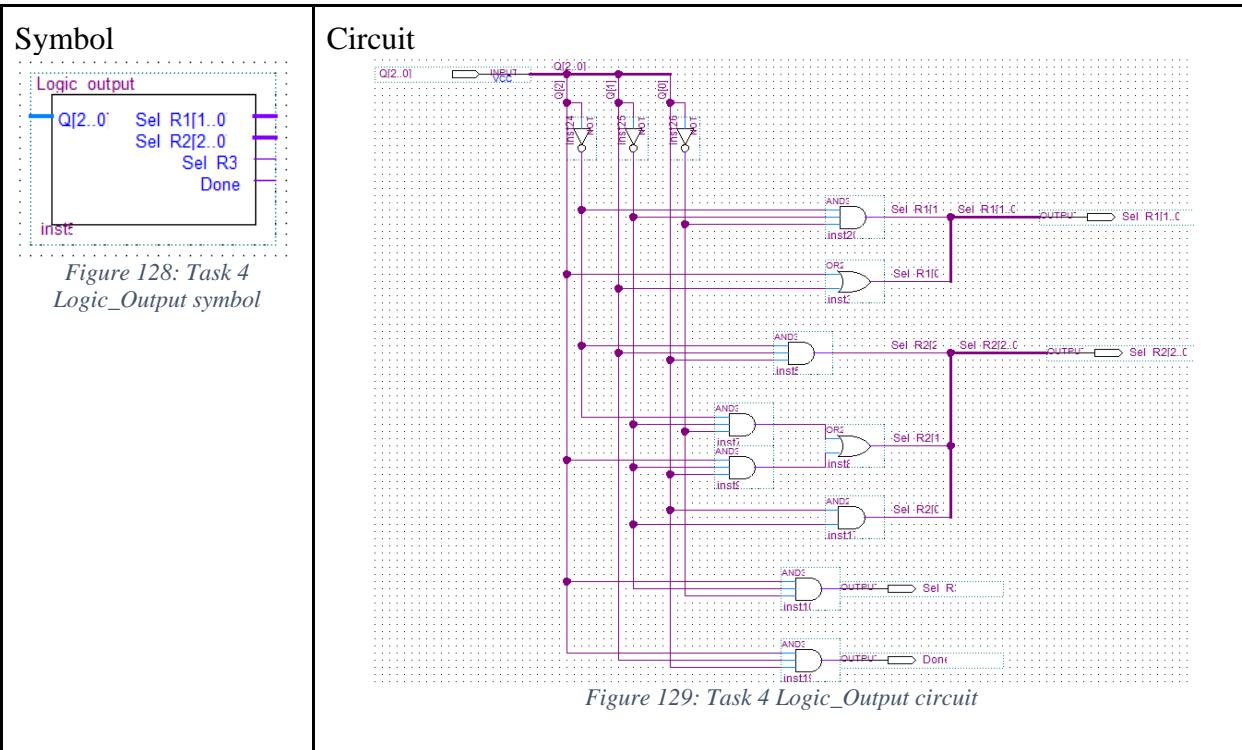
$$Q1+ = Q1'.Q0 + Q1.Q0'$$

$$Q0+ = Q0'.Start + Q1.Q0' + Q2.Q0'$$

### 4.3.2. Present\_state



### 4.3.3. Logic\_output



#### Logic\_output equation

$$\text{Sel\_ALU1} = Q1.Q0'$$

$$\text{Bus2} = Q2 + Q1 + Q0$$

$$\text{Sel\_ALU2}[1] = Q2.Q1'$$

$$\text{Bus3}[2] = Q1.Q0' + Q2.Q1 + Q2'.Q1'.Q0$$

$$\text{Sel\_ALU2}[0] = Q2'.Q1.Q0' + Q2.Q1'.Q0'$$

$$\text{Bus3}[1] = Q2'.Q1'.Q0'$$

$$\text{En\_R1} = Q2'.Q1' + Q1.Q0'$$

$$\text{Bus3}[0] = Q2'.Q1.Q0$$

$$\text{En\_R3} = Q2'.Q1.Q0$$

$$\text{Bus4}[2] = Q1'.Q0 + Q2.Q1' + Q2'.Q1.Q0'$$

$$\text{Bus1}[1] = Q1 + Q2'.Q0 + Q2.Q0'$$

$$\text{Bus4}[1] = Q2'.Q1'.Q0'$$

$$\text{Bus1}[0] = Q2.Q1'.Q0$$

$$\text{Bus4}[0] = Q2'.Q1.Q0$$

## 4.4. Waveform simulation

With 4 case: (-6, -8), (3, 4), (12, -5) and (24, -7)

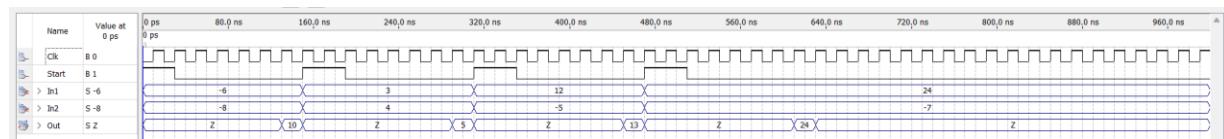


Figure 130: Task 4 Waveform

## Task 5: SRA design with Functional-unit pipelining

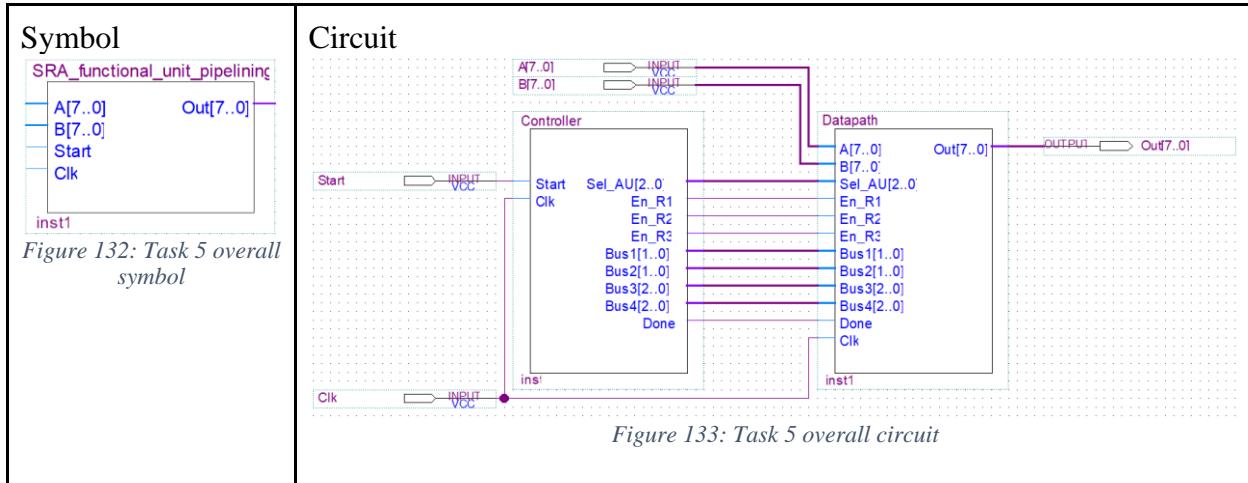
### Basic theory:

It takes 13 cycles to complete the calculation. However, one cycle will only be  $\frac{1}{2}$  of the longest processing time of the AU computation step (since the AU computation step is divided into 2 stages). According to the example in the previous slide note, 1 cycle is now only 50 ns. Total cost  $13 \times 50 = 650$  ns to complete the calculation.

	$s_0$	$s_1$	$s_2$	$s_3$	$s_4$	$s_5$	$s_6$	$s_7$	$s_8$	$s_9$	$s_{10}$	$s_{11}$	$s_{12}$
<b>Read <math>R_1</math></b>		a			$t_1$	$t_1$	x			x			$t_7$
<b>Read <math>R_2</math></b>			b		$t_2$	$t_2$	$t_3$		$t_5$		$t_6$		
<b>Read <math>R_3</math></b>									$t_4$				
<b>AU stage 1</b>		a	b		max	min	-				max		
<b>AU stage 2</b>			a	b		max	min	-				max	
<b>shifters</b>							>>3	>>1					
<b>Write <math>R_1</math></b>	a		$t_1$			x		$t_5$					$t_7$
<b>Write <math>R_2</math></b>	b			$t_2$		$t_3$			$t_6$				
<b>Write <math>R_3</math></b>								$t_4$					
<b>Outport</b>													$t_7$

Figure 131: Task 5 timing diagram

### 5.1. Overall Design



## 5.2. Datapath Design

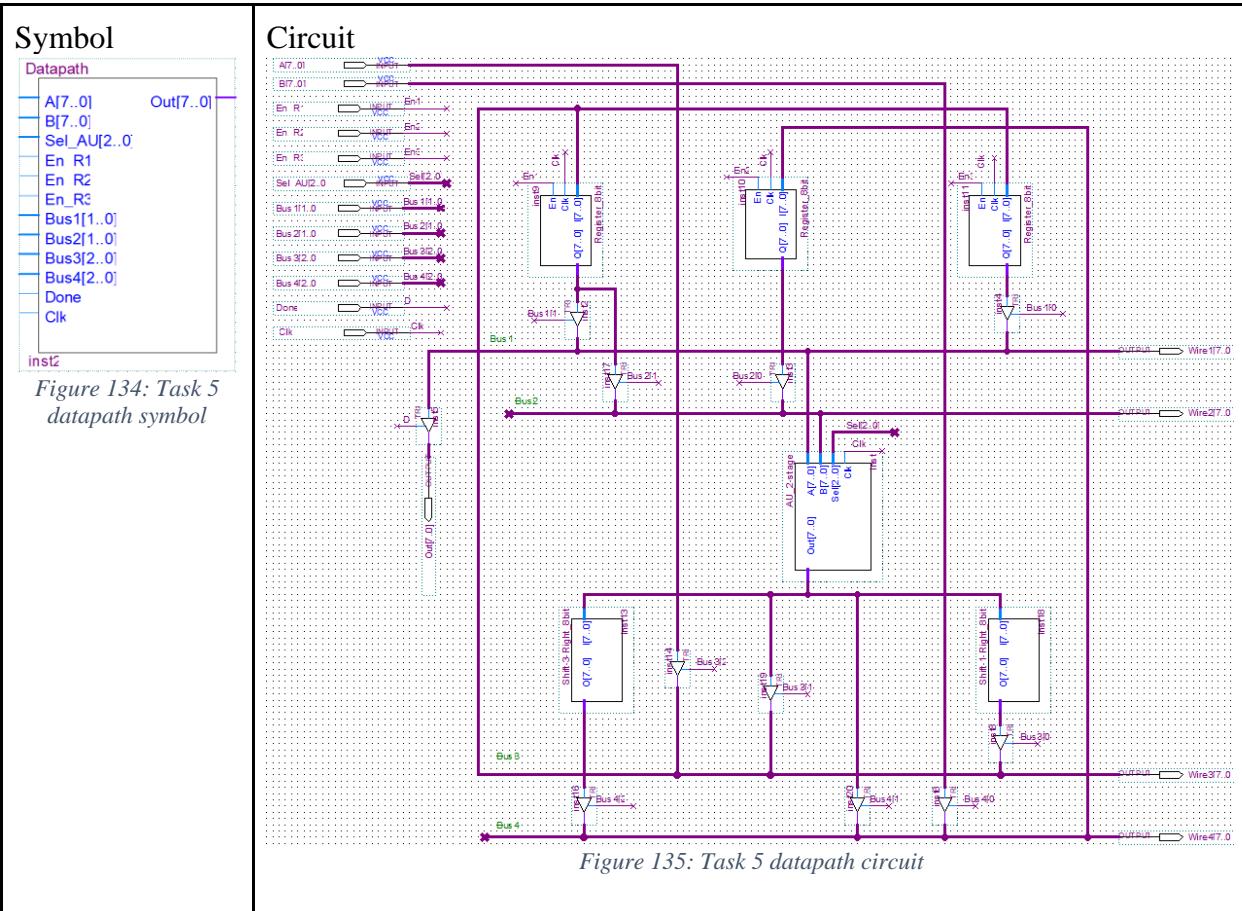


Figure 134: Task 5 datapath symbol

Figure 135: Task 5 datapath circuit

### 5.2.1. Register\_8bit

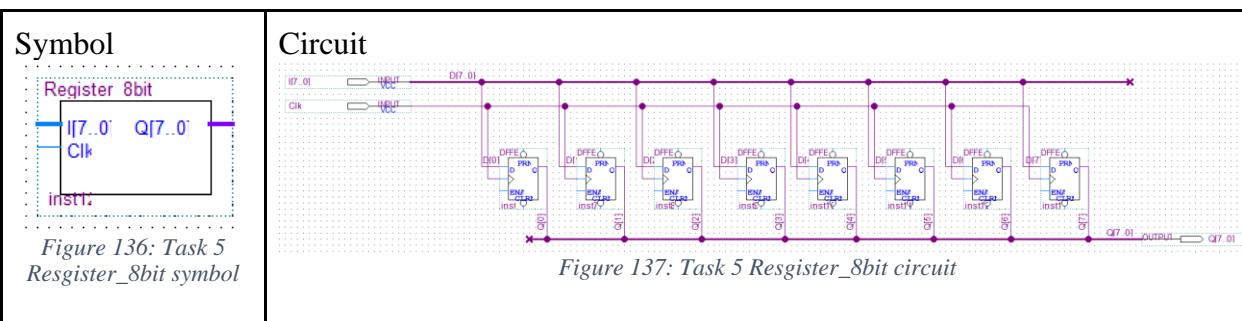
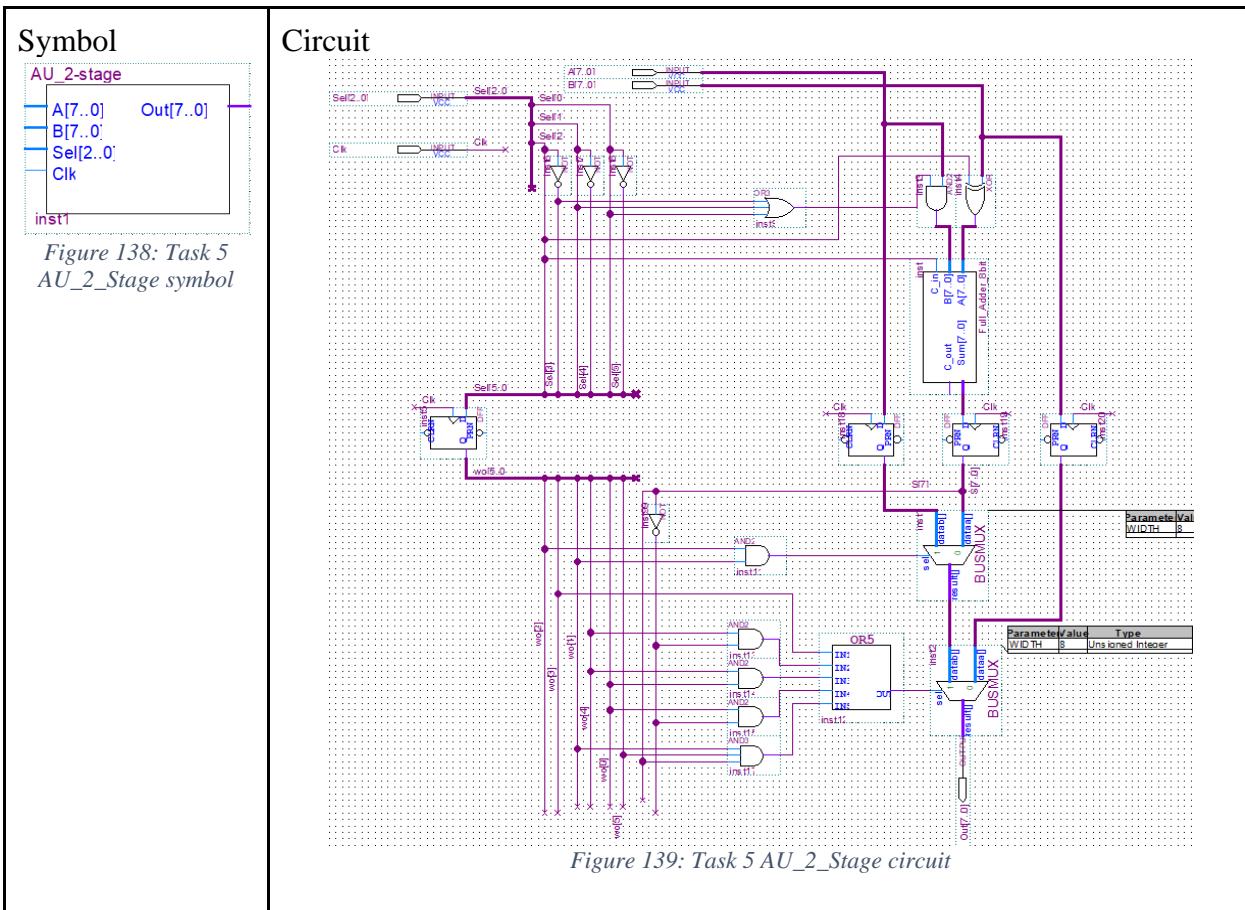


Figure 136: Task 5 Register\_8bit symbol

Figure 137: Task 5 Register\_8bit circuit

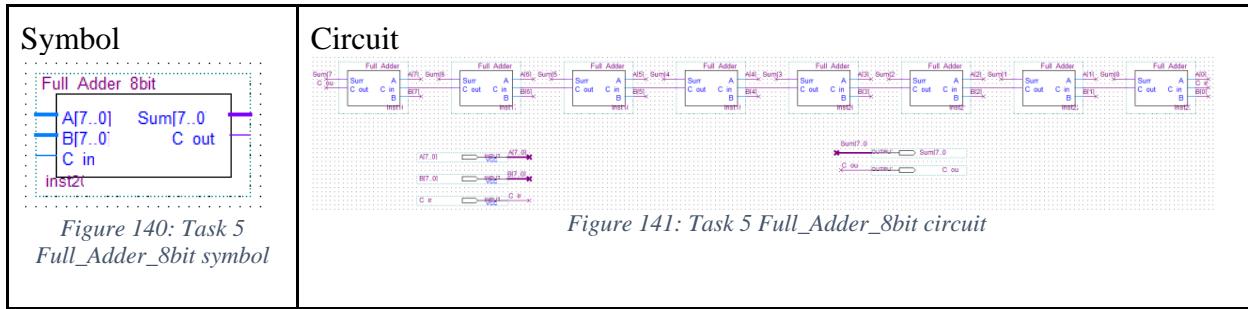
### 5.2.2. AU-2-stage



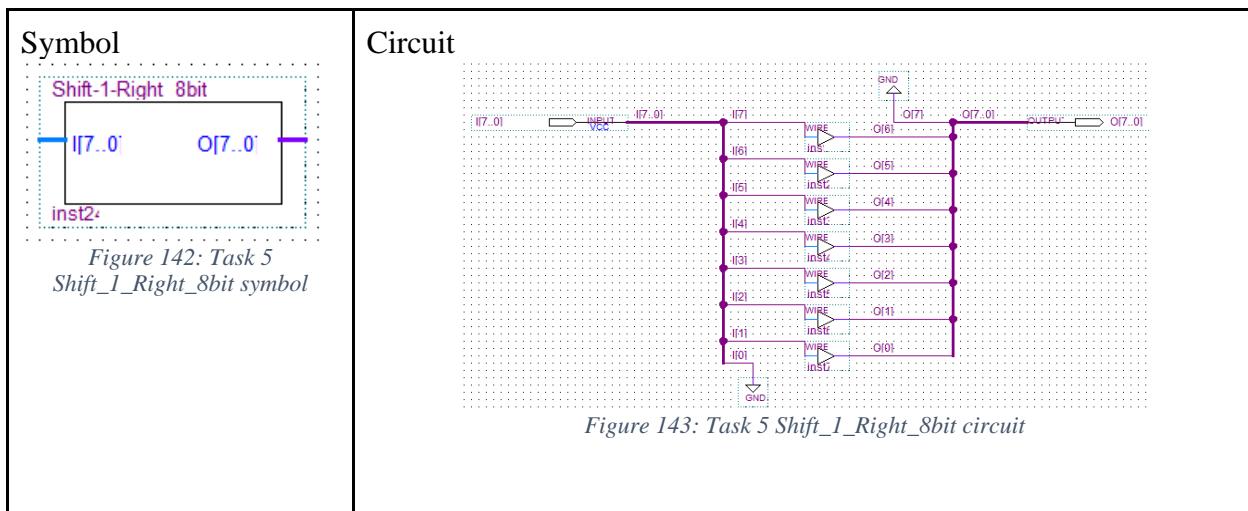
Operation table

Sel2	Sel1	Sel0	Function
0	0	0	Add
0	0	1	
0	1	0	
0	1	1	
1	0	0	Abs
1	0	1	Sub
1	1	0	Min
1	1	1	Max

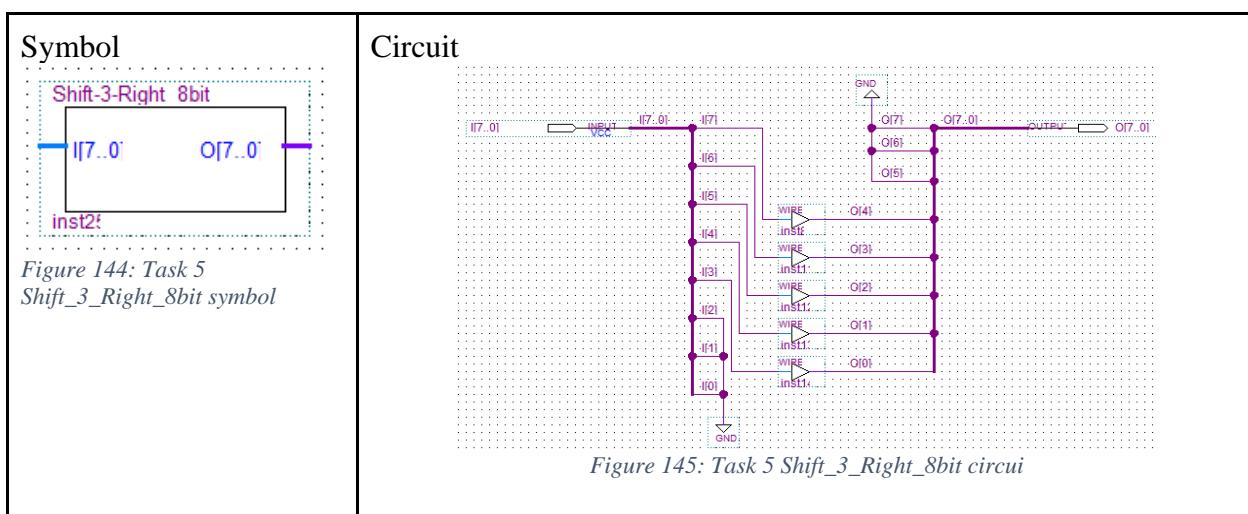
### 5.2.3. Full\_Adder\_8bit



### 5.2.4. Shift-1-Right\_8bit



### 5.2.5. Shift-3-Right\_8bit



### 5.3. Controller Design

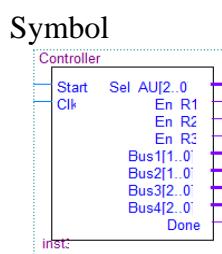


Figure 146: Task 5 controller symbol

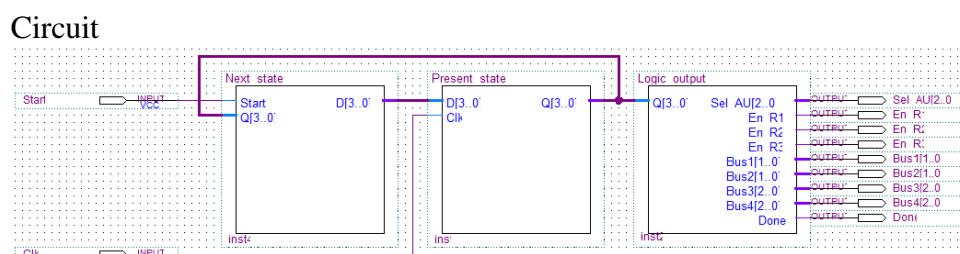


Figure 147: Task 5 controller circuit

State Table

	Q3Q2Q1Q0	Start	Q3+Q2+Q1+Q0+	Sel_ALU[3]	En_R1	En_R2	En_R3	Bus1[2]	Bus2[2]	Bus3[3]	Bus4[3]
S0	0000	0	0000	100	1	1	0	00	00	100	001
	0000	1	0001	100	1	1	0	00	00	100	001
S1	0001	0	0010	100	0	0	0	00	10	010	000
	0001	1	0010	100	0	0	0	00	10	010	000
S2	0010	0	0011	100	1	0	0	00	01	010	010
	0010	1	0011	100	1	0	0	00	01	010	010
S3	0011	0	0100	100	0	1	0	00	01	000	010
	0011	1	0100	100	0	1	0	00	01	000	010
S4	0100	0	0101	111	0	0	0	10	01	000	000
	0100	1	0101	111	0	0	0	10	01	000	000
S5	0101	0	0110	110	1	1	0	10	01	010	100
	0101	1	0110	110	1	1	0	10	01	010	100
S6	0110	0	0111	101	0	0	1	10	01	001	000
	0110	1	0111	101	0	0	1	10	01	001	000
S7	0111	0	1000	101	0	1	0	10	01	010	010
	0111	1	1000	101	0	1	0	10	01	010	010
S8	1000	0	1001	OXX	0	0	0	01	01	010	010
	1000	1	1001	OXX	0	0	0	01	01	010	010
S9	1001	0	1010	OXX	0	1	0	01	01	010	010
	1001	1	1010	OXX	0	1	0	01	01	010	010
S10	1010	0	1011	111	0	0	0	10	01	010	000
	1010	1	1011	111	0	0	0	10	01	010	000
S11	1011	0	1100	111	1	0	0	10	01	010	000
	1011	1	1100	111	1	0	0	10	01	010	000
S12	1100	0	0000	111	0	0	0	10	00	000	000
	1100	1	0000	111	0	0	0	10	00	000	000

State Diagram

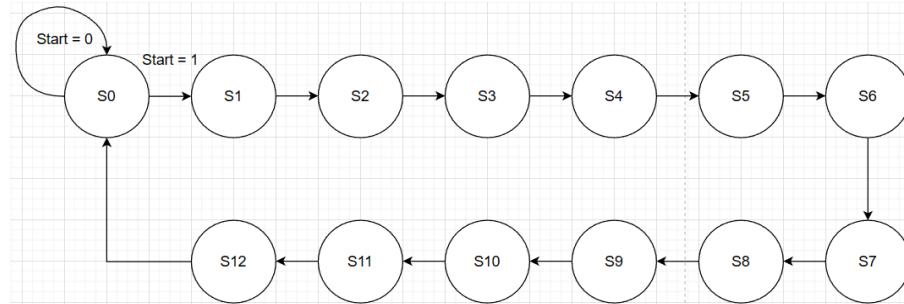


Figure 148: Task 5 state diagram

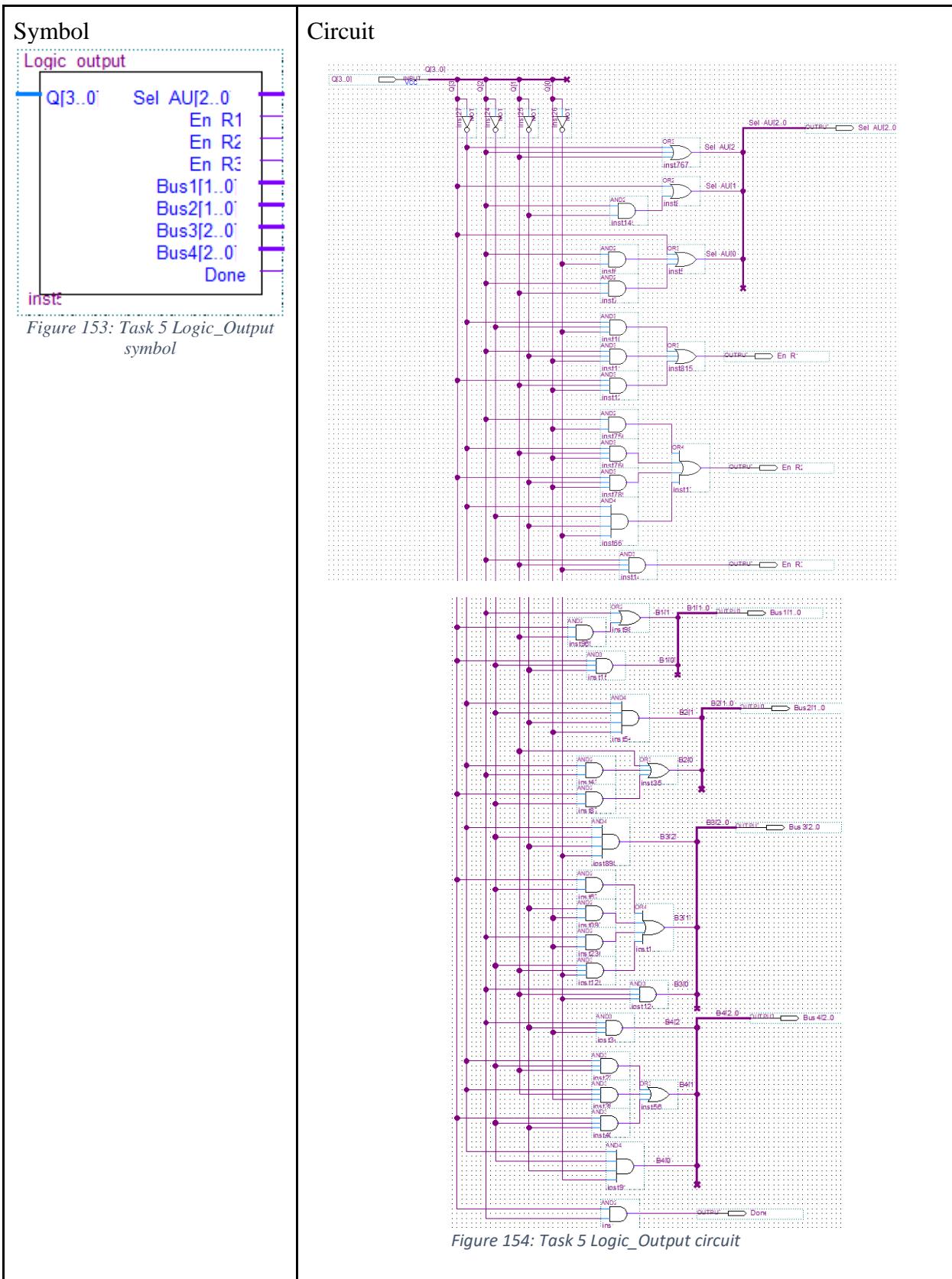
### 5.3.1. Next\_state

<p><b>Symbol</b></p> <p>Figure 149: Task 5 Next_State symbol</p>	<p><b>Circuit</b></p> <p>Figure 150: Task 5 Next_State circuit</p>
<p><b>Next_state equation</b></p> $Q3+ = Q3.Q2' + Q2.Q1.Q0$ $Q2+ = Q2'.Q1.Q0 + Q3'.Q2.Q1' + Q3'.Q2.Q0'$ $Q1+ = Q1'.Q0 + Q1.Q0'$ $Q0+ = Q1.Q0' + Q3'.Q0'.Start + Q3'.Q2.Q0' + Q3.Q2'Q0'$	

### 5.3.2. Present\_state

<p><b>Symbol</b></p> <p>Figure 151: Task 5 Present_State symbol</p>	<p><b>Circuit</b></p> <p>Figure 152: Task 5 Present_State circuit</p>
---	---

### 5.3.3. Logic\_output



### Logic\_output equation

$$\text{Sel\_AU}[2] = Q3' + Q2 + Q1$$

$$\text{Sel\_AU}[1] = Q3 + Q2.Q1'$$

$$\text{Sel\_AU}[0] = Q3 + Q2.Q0' + Q2.Q1$$

$$\text{En\_R1} = Q3'.Q2'.Q0' + Q2.Q1'.Q0 + Q3.Q1.Q0$$

$$\text{En\_R2} = Q2.Q0 + Q3'.Q1.Q0 + Q3.Q1'.Q0 + Q3'.Q2'.Q1'.Q0'$$

$$\text{En\_R3} = Q2.Q1.Q0'$$

$$\text{Bus1}[1] = Q2 + Q3.Q1$$

$$\text{Bus1}[0] = Q3.Q2'.Q1'$$

$$\text{Bus2}[1] = Q3'.Q2'.Q1'.Q0$$

$$\text{Bus2}[0] = Q1 + Q3'.Q2 + Q3.Q2'$$

$$\text{Bus3}[2] = Q3'.Q2'.Q1'.Q0'$$

$$\text{Bus3}[1] = Q3.Q2' + Q1'.Q0 + Q2.Q0 + Q2'.Q1.Q0'$$

$$\text{Bus3}[0] = Q2.Q1.Q0'$$

$$\text{Bus4}[2] = Q2.Q1'.Q0$$

$$\text{Bus4}[1] = Q3'.Q2'.Q1 + Q3'.Q1.Q0 + Q3.Q2'.Q1'$$

$$\text{Bus4}[0] = Q3'.Q2'.Q1'.Q0'$$

$$\text{DONE} = Q3.Q2$$

## 5.4. Waveform simulation

With 3 case: (-3, 4), (-6, 8) and (15, -8)

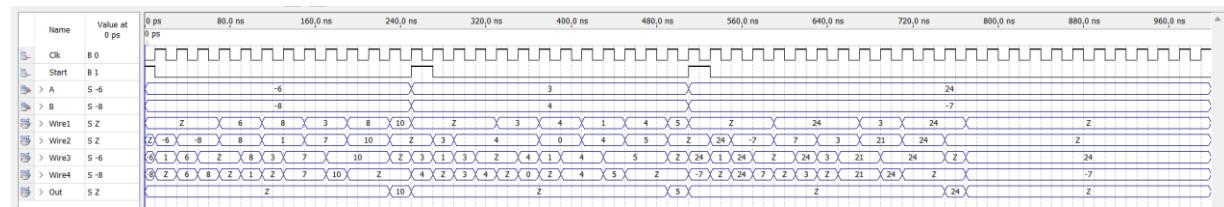


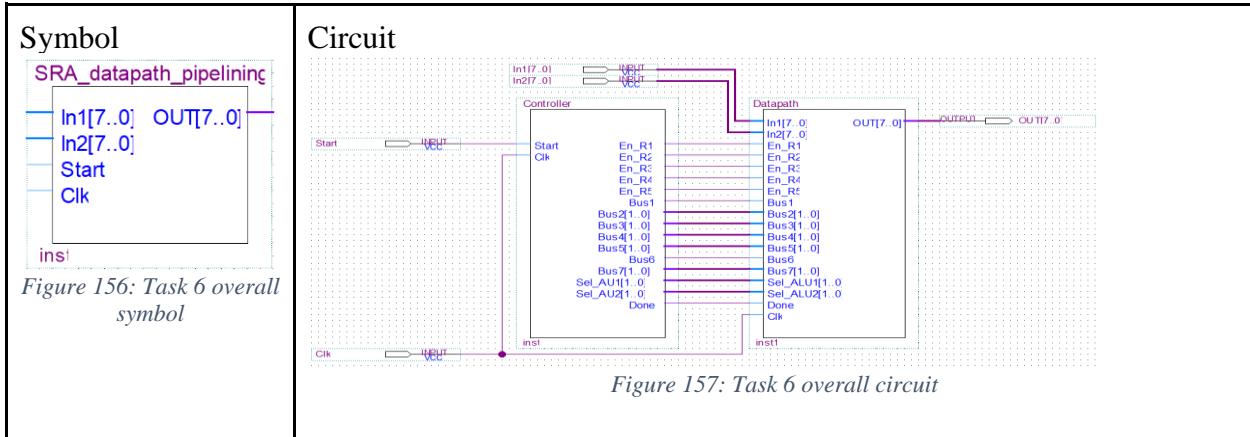
Figure 155: Task 5 waveform

## Task 6: SRA design with Datapath pipelining

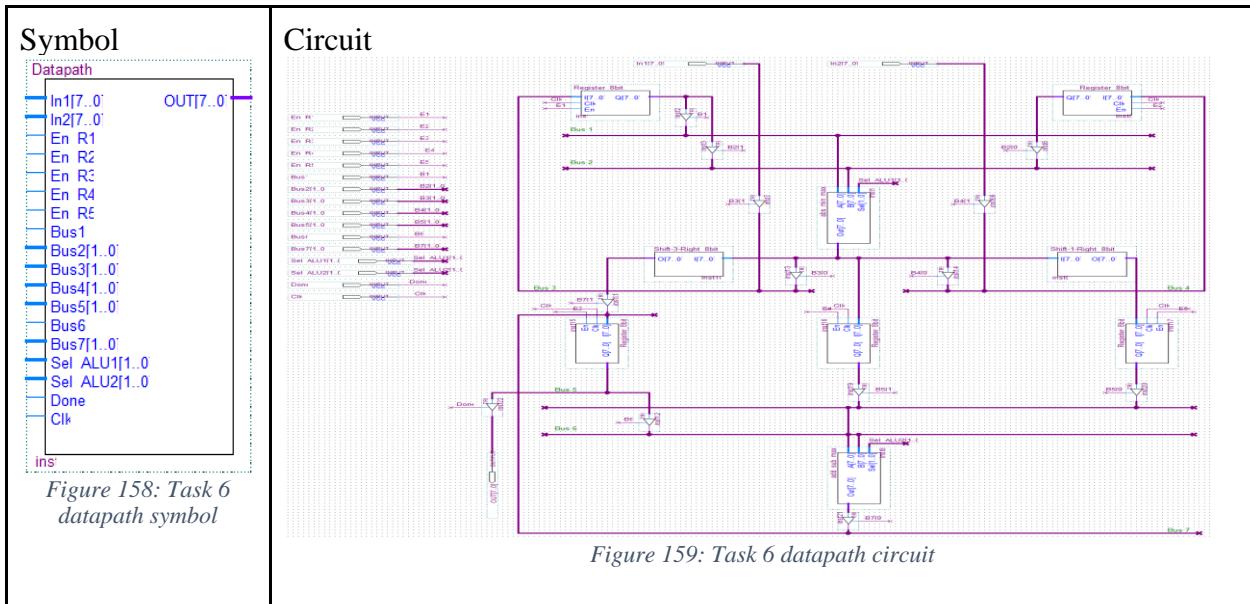
### Basic theory:

In case the data line block performs the same function for many different input signal sets, we can further improve it by designing a pipeline for the entire ASM graph. To do this, we split the entire ASM graph into equal-sized chunks and then use different data line layers to execute each split. By this design, all layers can execute with the input operand sets concurrently, each layer produces a partial result and this result is used by the succeeding datapath layers.

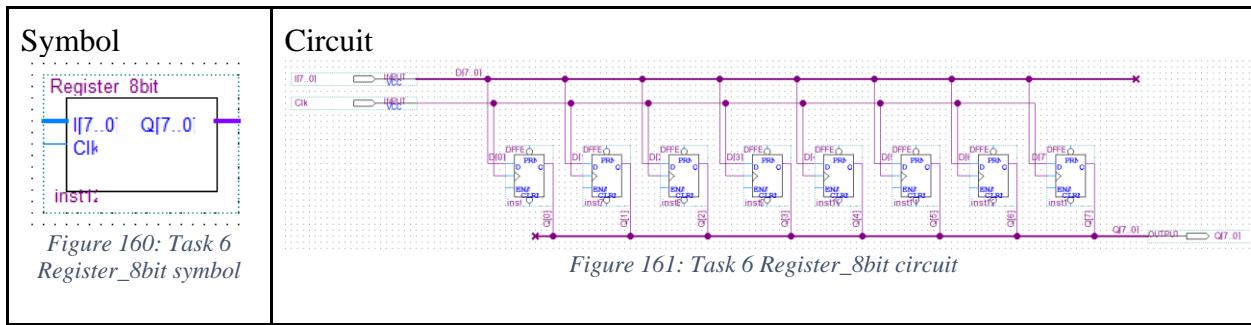
### 6.1. Overall Design



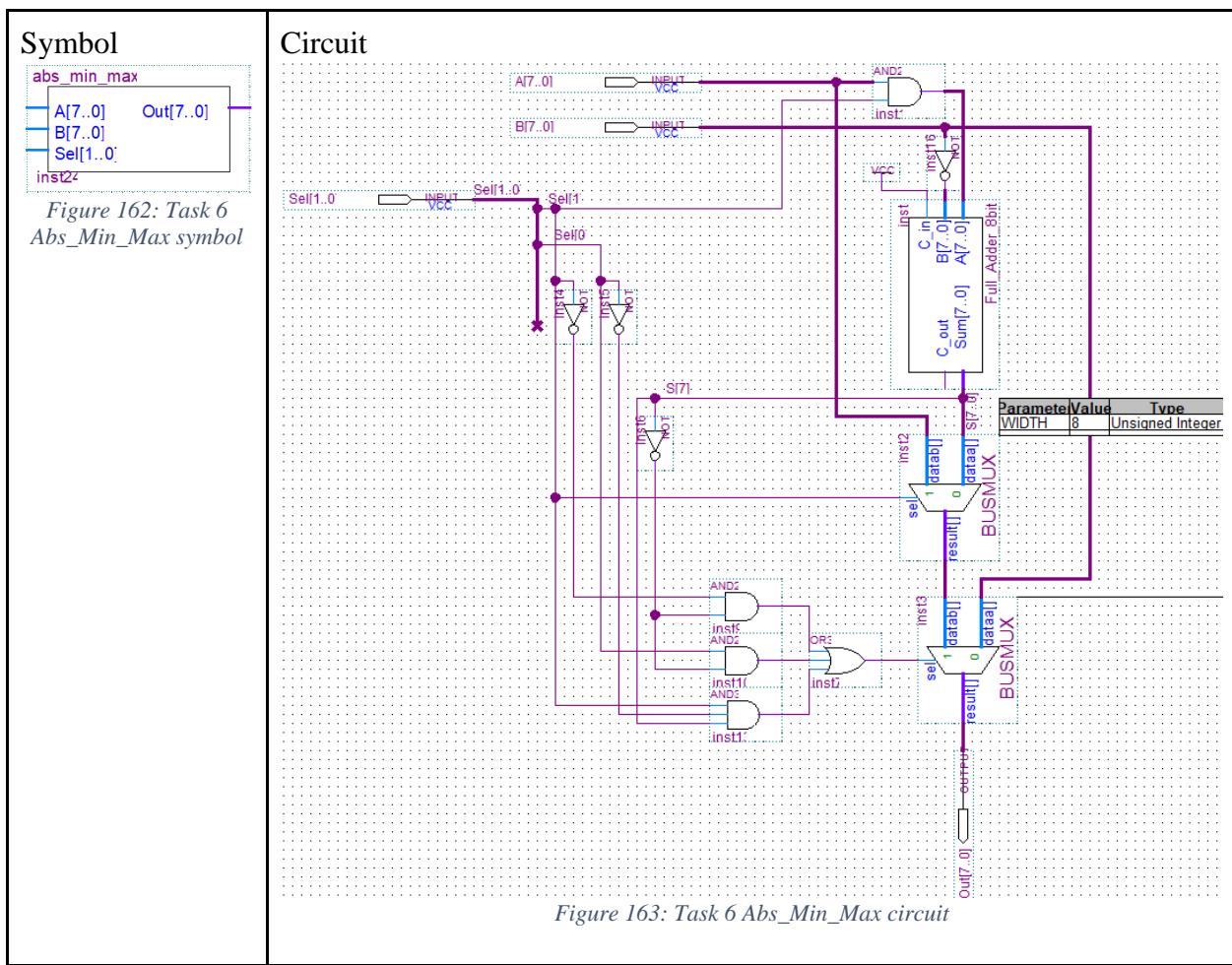
### 6.2. Datapath Design



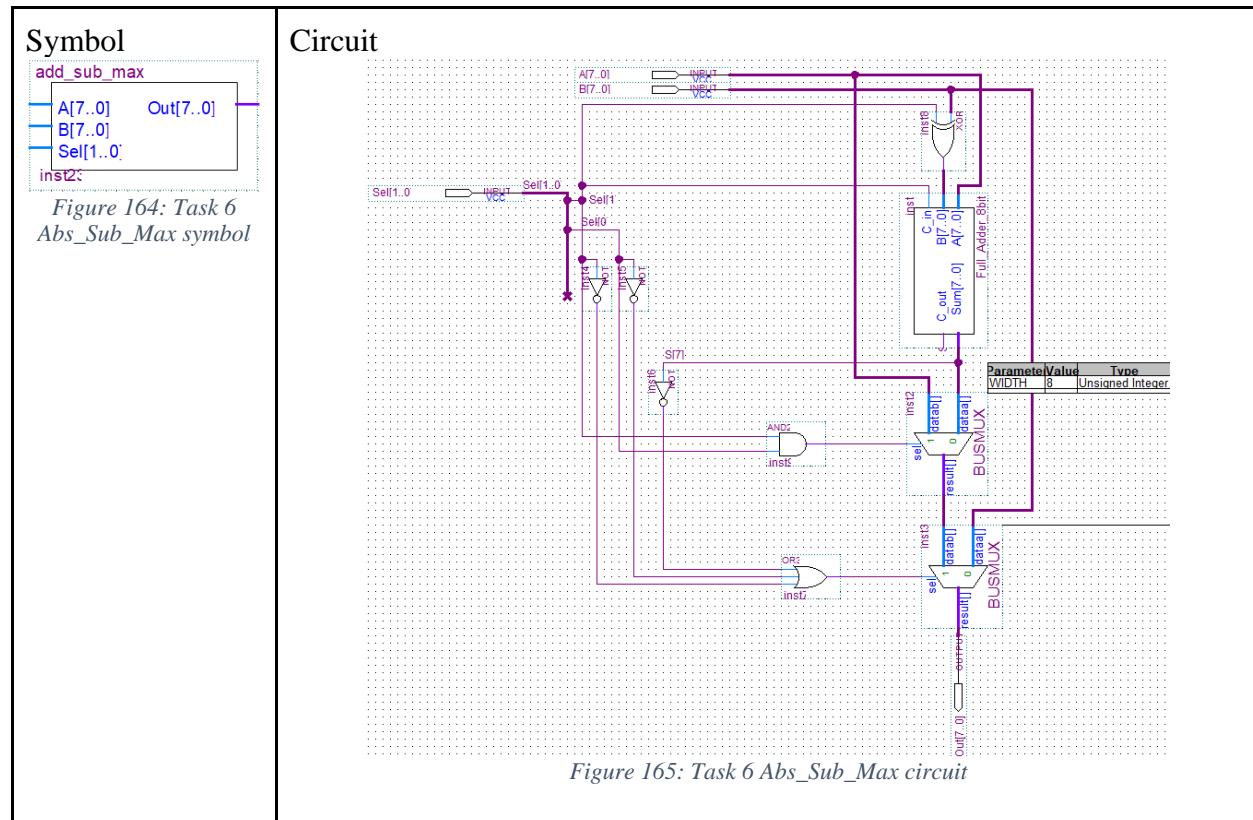
### 6.2.1. Register\_8bit



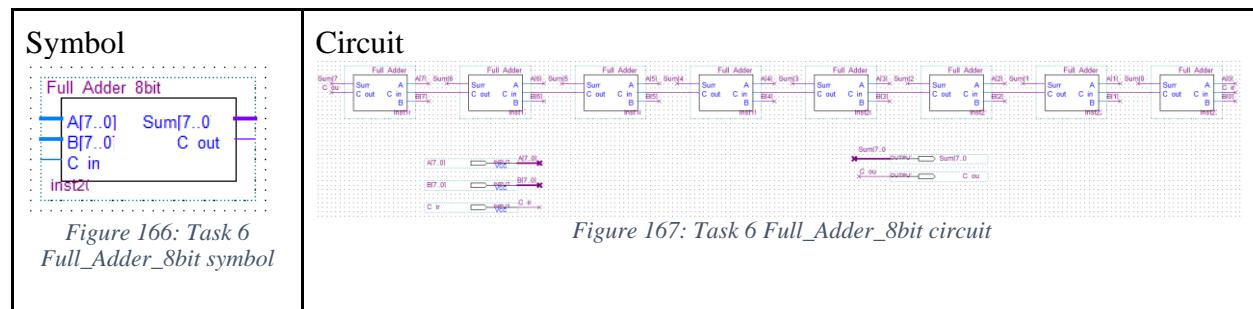
### 6.2.2. abs\_min\_max



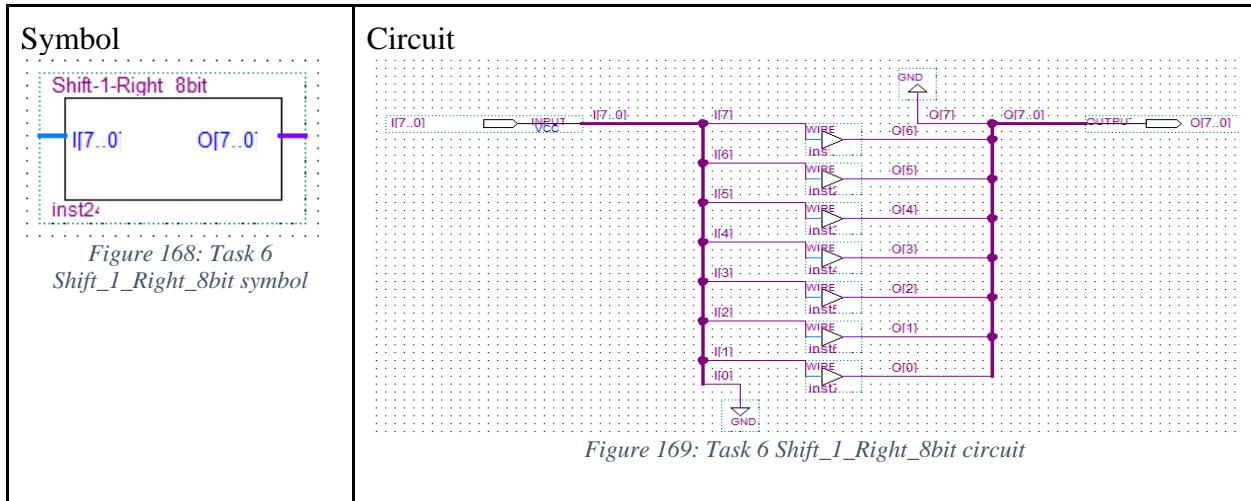
### 6.2.3. add\_sub\_max



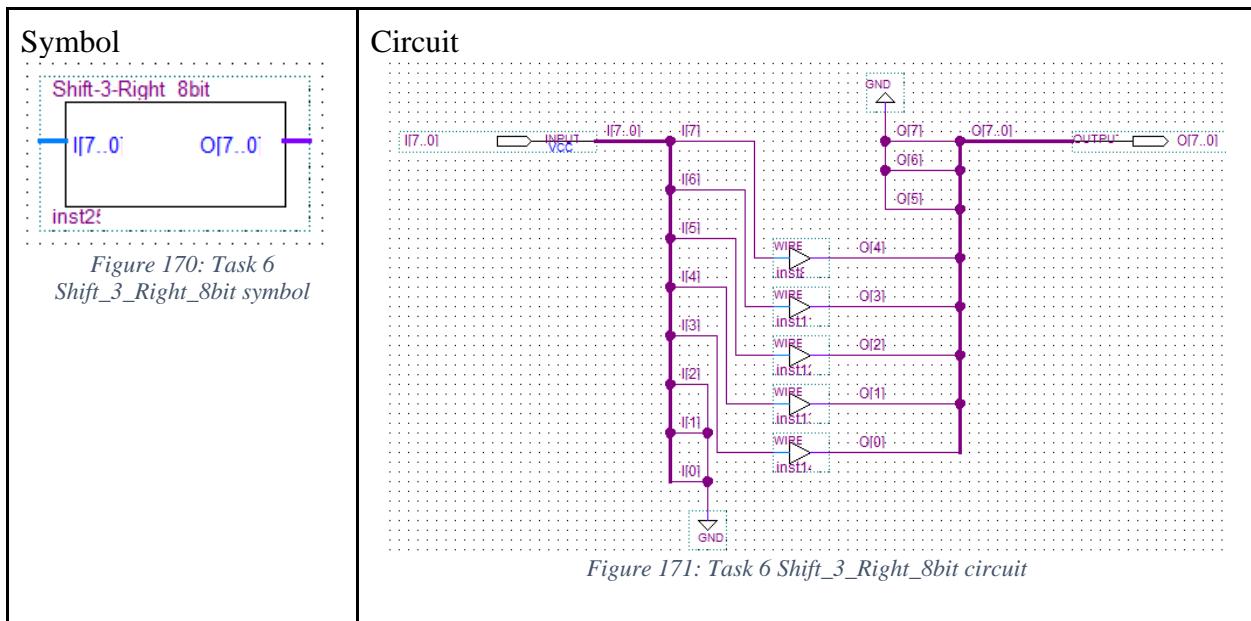
### 6.2.4. Full\_Adder\_8bit



### 6.2.5. Shift-1-Right\_8bit



### 6.2.6. Shift-3-Right\_8bit



### 6.3. Controller Design

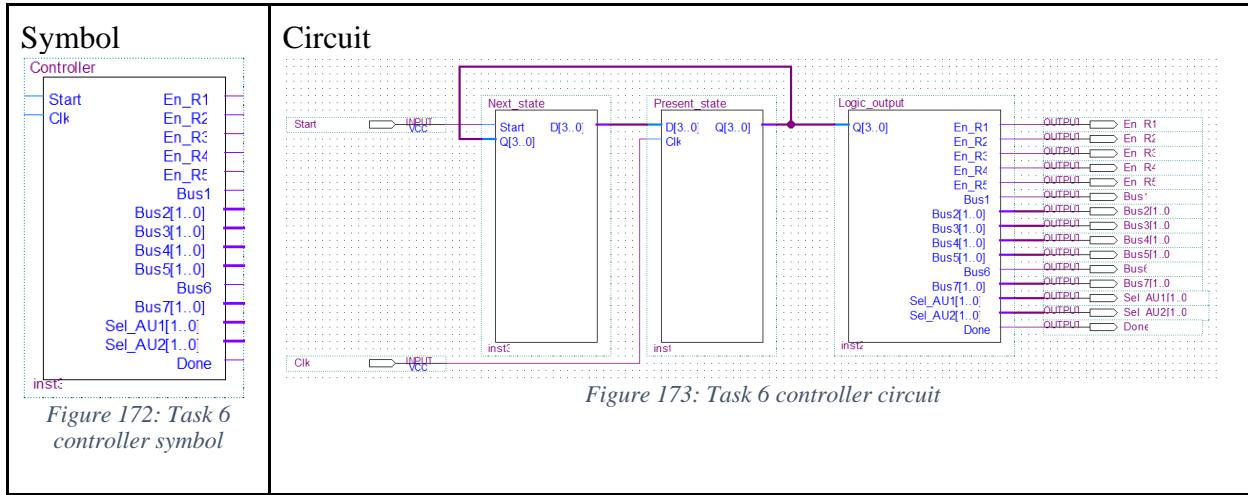


Figure 173: Task 6 controller circuit

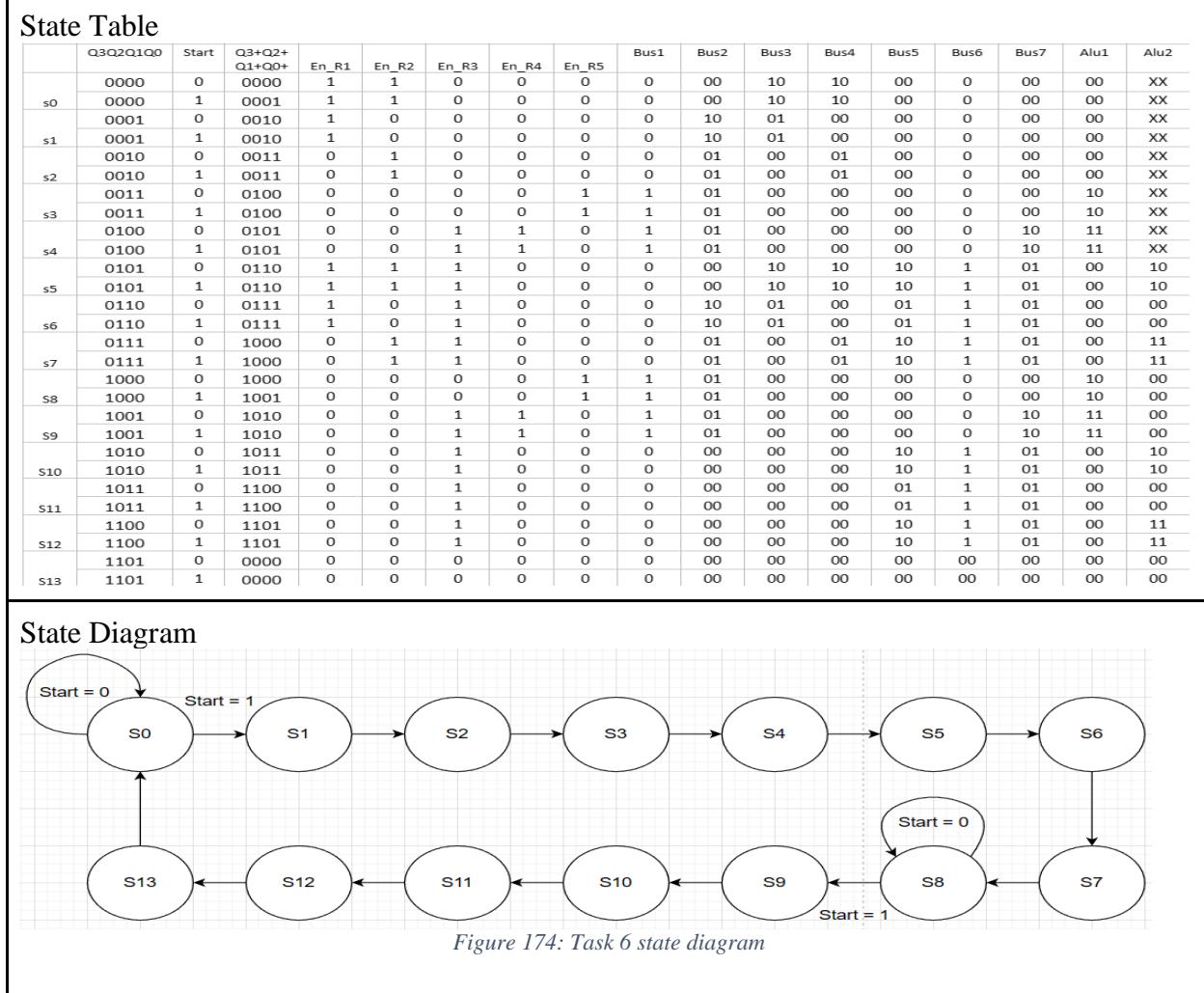
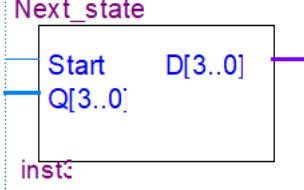
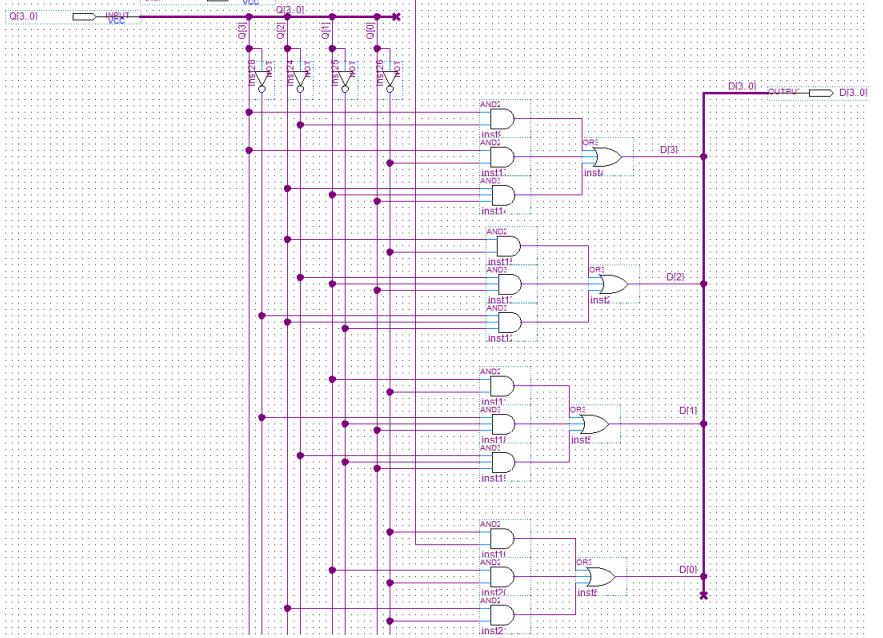


Figure 174: Task 6 state diagram

### 6.3.1. Next\_state

Symbol	Circuit
<p><b>Next_state</b></p>  <p>Figure 175: Task 6 Next_State symbol</p>	 <p>Figure 176: Task 6 Next_State circuit</p>

Next\_state equation

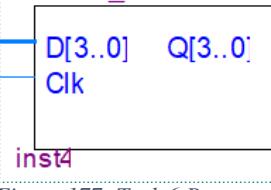
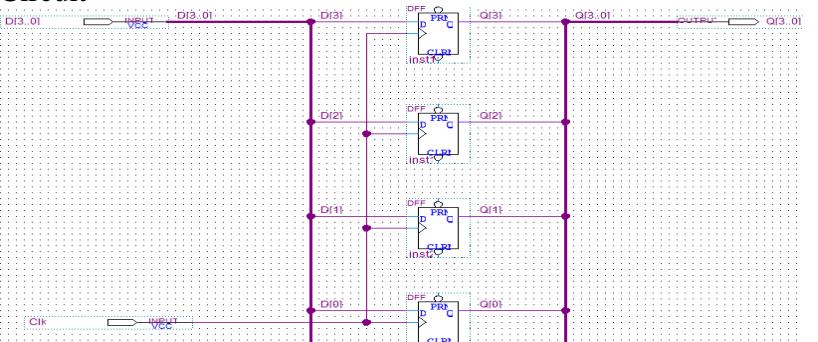
$$Q3+ = Q3.Q2' + Q3.Q0' + Q2.Q1.Q0$$

$$Q2+ = Q2.Q0' + Q2'.Q1.Q0 + Q3'.Q2.Q1$$

$$Q1+ = Q1.Q0' + Q3'.Q1'.Q0 + Q2'.Q1'.Q0$$

$$Q0+ = Q0'.Start + Q1.Q0' + Q2.Q0'$$

### 6.3.2. Present\_state

Symbol	Circuit
<p><b>Present_state</b></p>  <p>Figure 177: Task 6 Present_State symbol</p>	 <p>Figure 178: Task 6 Present_State circuit</p>

### 6.3.3. Logic\_output

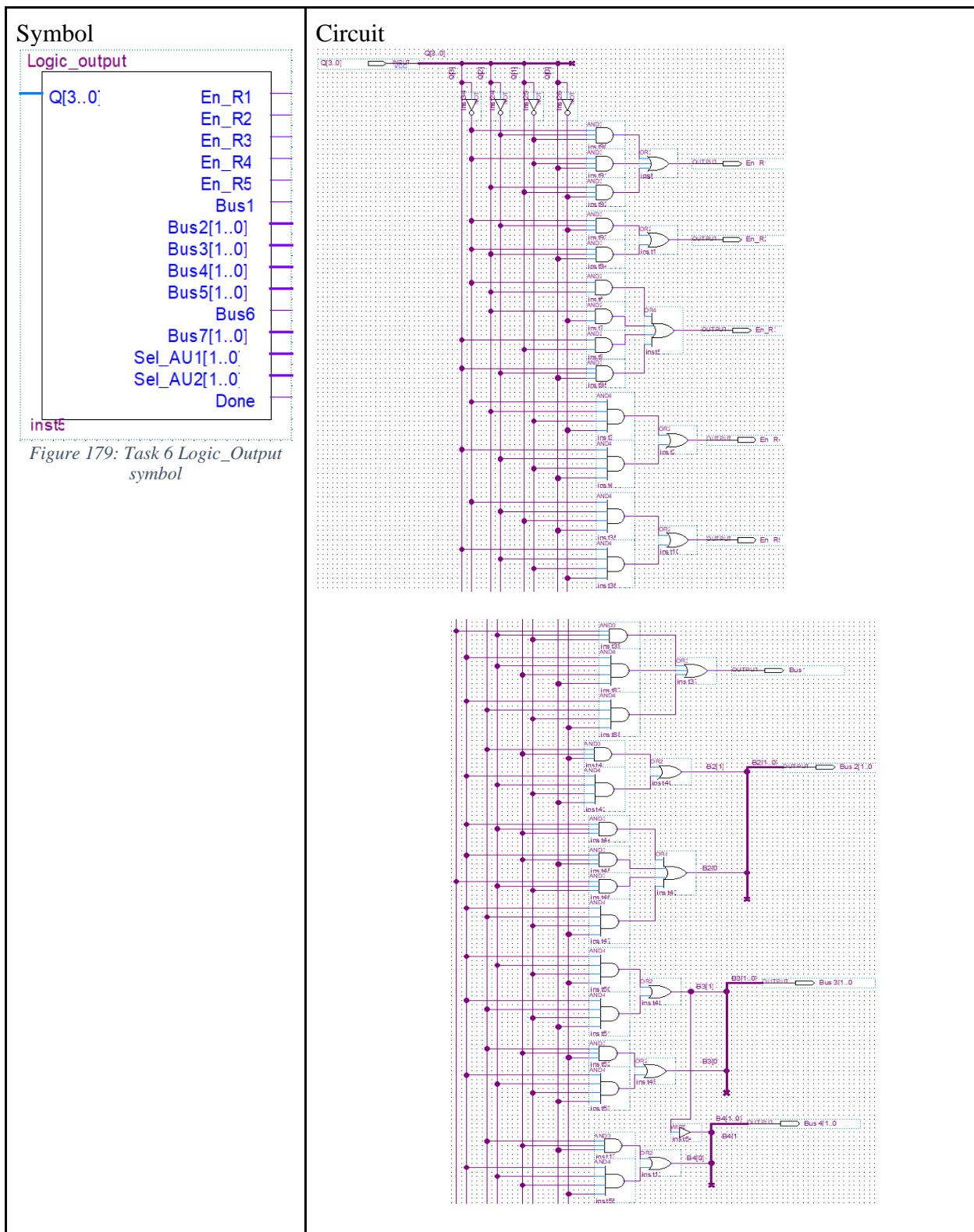


Figure 179: Task 6 Logic\_Output symbol

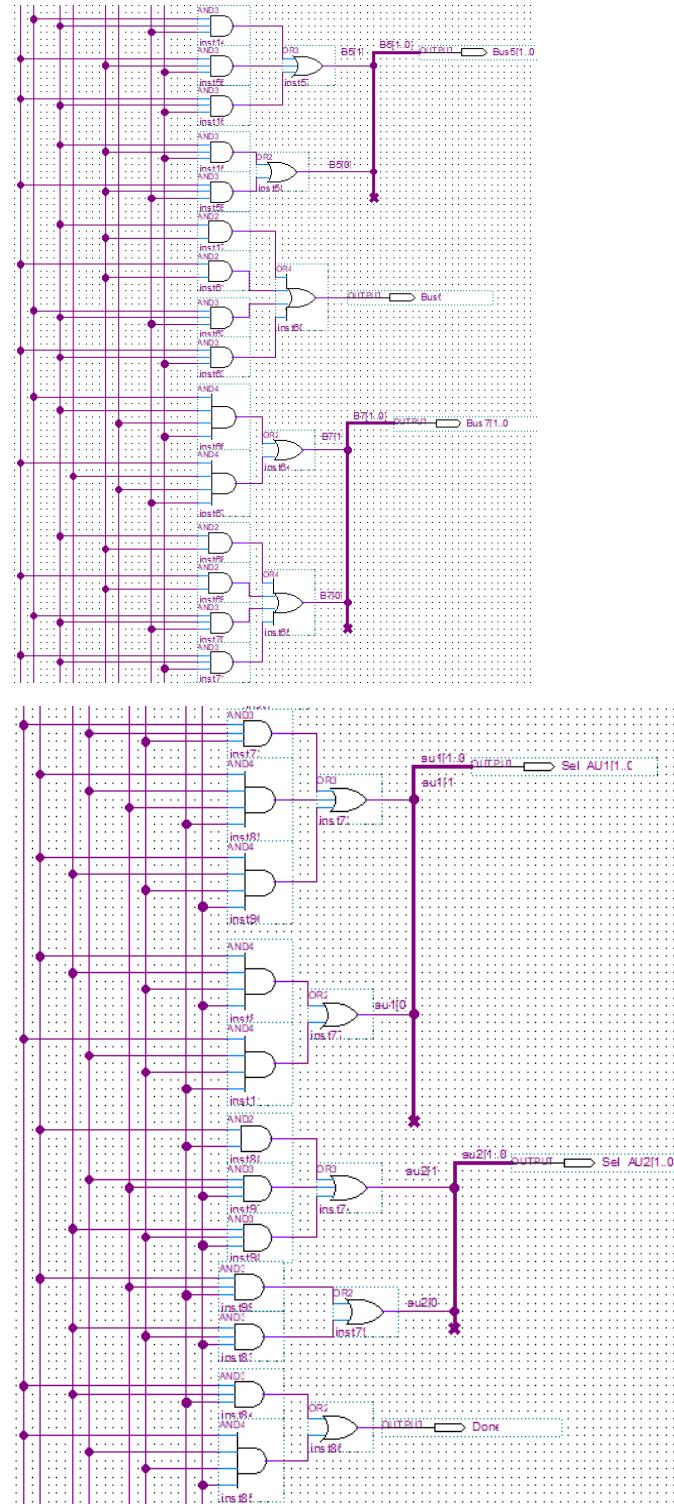


Figure 180: Task 6 Logic\_Output circuit

### Logic\_output equation

$$En\_R1 = Q3' \cdot Q2' \cdot Q1' + Q3' \cdot Q1' \cdot Q0 + Q2 \cdot Q1 \cdot Q0'$$

```

En_R2 = Q3'.Q2'.Q0' + Q3'.Q2.Q0
En_R3 = Q3'Q2 + Q2.Q0' + Q3.Q1 + Q3.Q2'.Q0
En_R4 = Q3'.Q2.Q1'.Q0' + Q3.Q2'.Q1'.Q0
En_R5 = Q3'.Q2'.Q1.Q0 + Q3.Q2'.Q1'.Q0'
Bus1 = Q3.Q2'.Q1' + Q3'.Q2'.Q1.Q0 + Q3'.Q2.Q1'.Q0'
Bus2[1] = Q2.Q1.Q0' + Q3'.Q2'.Q1'.Q0
Bus2[0] = Q3'.Q2'.Q1 + Q3'.Q1.Q0 + Q3.Q2'.Q1' + Q3'.Q2.Q1'.Q0'
Bus3[1] = Q3'.Q2'.Q1'.Q0' + Q3'.Q2.Q1'.Q0
Bus3[0] = Q2.Q1.Q0' + Q3'.Q2'.Q1'.Q0
Bus4[1] = Q3'.Q2'.Q1'.Q0' + Q3'.Q2.Q1'.Q0
Bus4[0] = Q2.Q1.Q0 + Q3'.Q2'.Q1.Q0'
Bus5[1] = Q3'.Q2.Q0 + Q3.Q1.Q0' + Q3.Q2.Q0'
Bus5[0] = Q2.Q1.Q0' + Q3.Q1.Q0
Bus6 = Q2.Q1 + Q3.Q1 + Q3'.Q2.Q0 + Q3.Q2.Q0'
Bus7[1] = Q3'.Q2.Q1'.Q0' + Q3.Q2'.Q1'.Q0
Bus7[0] = Q2.Q1 + Q3.Q1 + Q3'.Q2.Q0 + Q3.Q2.Q0'
Sel_AU1[1] = Q3.Q2'.Q1' + Q3.'Q2'.Q1.Q0 + Q3'.Q2.Q1'.Q0'
Sel_AU1[0] = Q3'.Q2.Q1'.Q0' + Q3.Q2'.Q1'.Q0
Sel_AU2[1] = Q3'.Q0 + Q2'.Q1.Q0' + Q2.Q1'.Q0'
Sel_AU2[0] = Q3'.Q1.Q0 + Q2.Q1'.Q0'

```

## 6.4. Waveform simulation

With 4 case: (-6, -8), (-3, 4), (12, 16) and (24, -32)

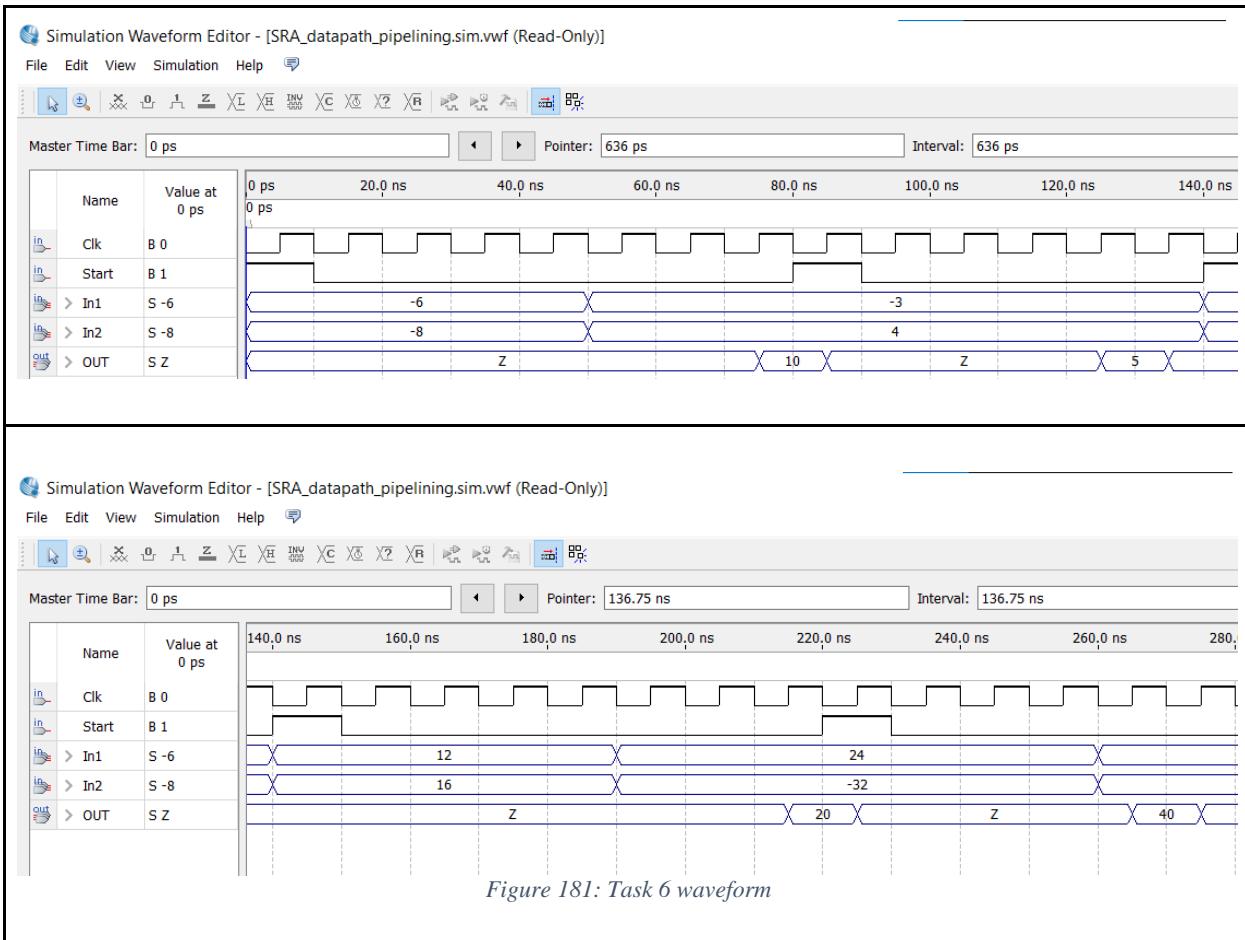


Figure 181: Task 6 waveform

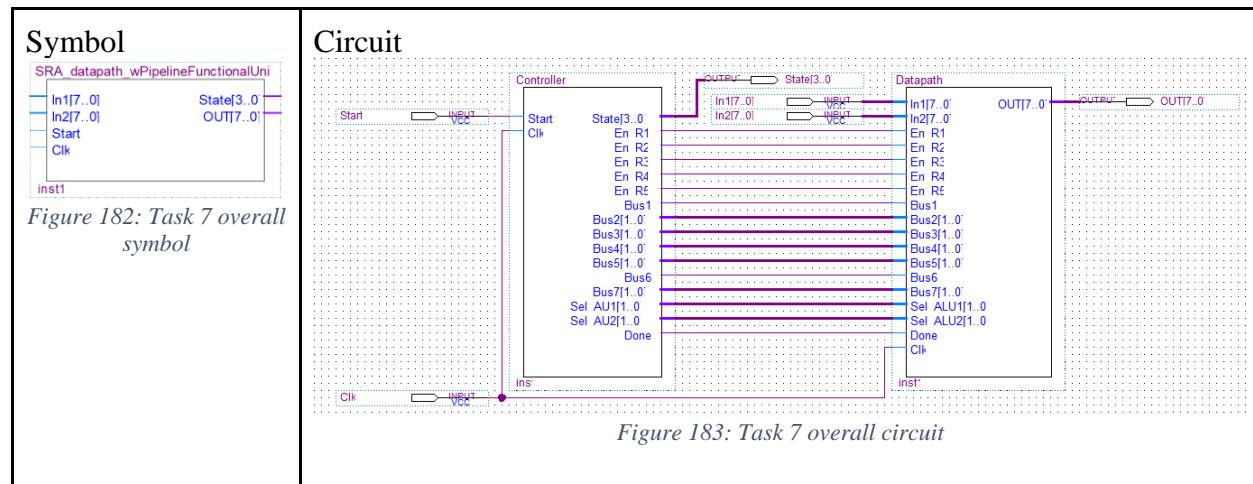
## Task 7: Datapath pipelining with Functional Unit Pipeline

### Basic theory:

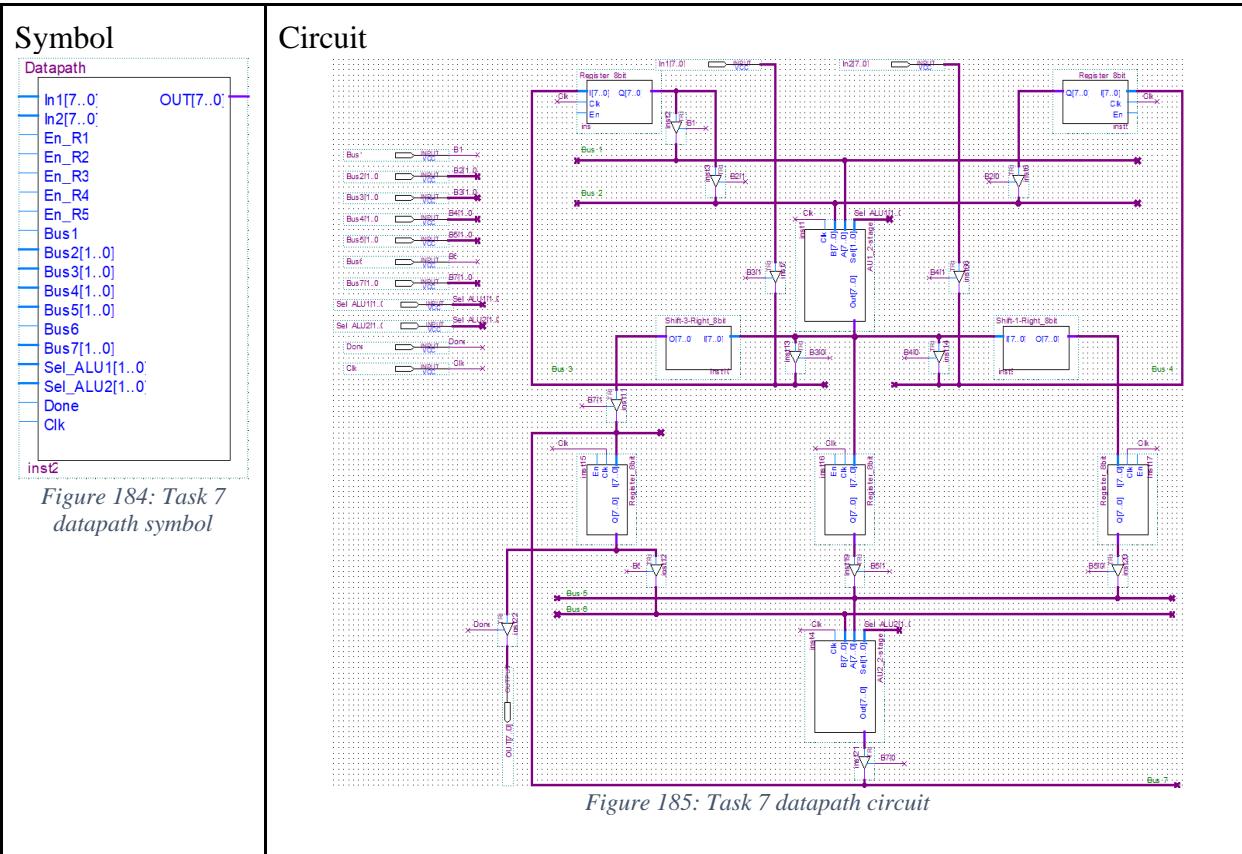
Combine both datapath pipeline and pipelined functional units, meaning that in the datapath pipeline (previous slide) the AU1, and AU2 in each stage-datapath pipeline will be divided into 2-stage functional units again. Each stage-datapath pipeline takes 7 cycles, but each cycle is reduced by  $\frac{1}{2}$ .

For m values to calculate RSA, it takes  $7m + 7$  cycles, but compared to the cycle in the datapath pipeline, the time per cycle in the datapath pipeline - pipelined functional units is reduced by  $\frac{1}{2}$

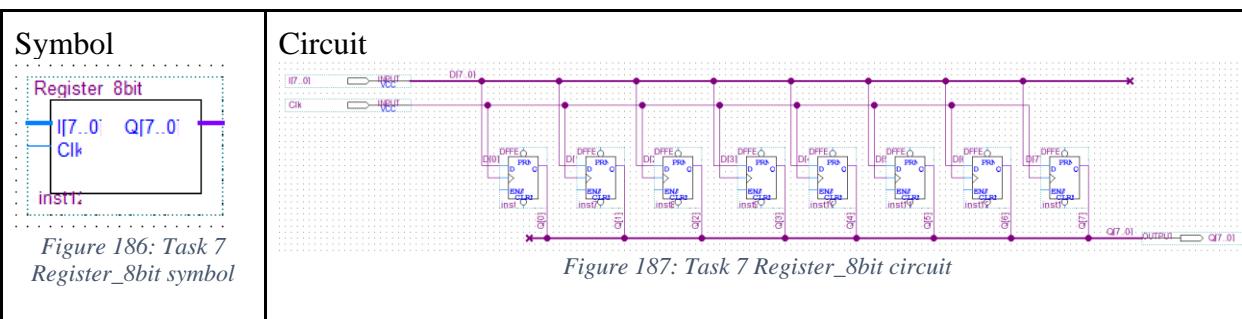
### 7.1. Overall Design



## 7.2. Datapath Design



### 7.2.1. Register\_8bit



### 7.2.2. AU1\_2-stage

Symbol

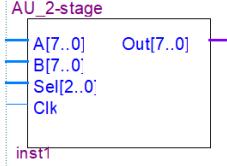


Figure 188: Task 7 AU1\_2Stage symbol

Circuit

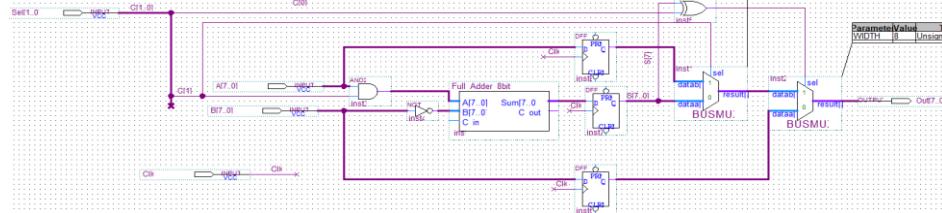


Figure 189: Task 7 AU1\_2Stage circuit

### 7.2.2. AU2\_2-stage

Symbol

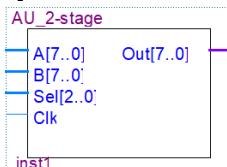


Figure 190: Task 7 AU2\_2Stage symbol

Circuit

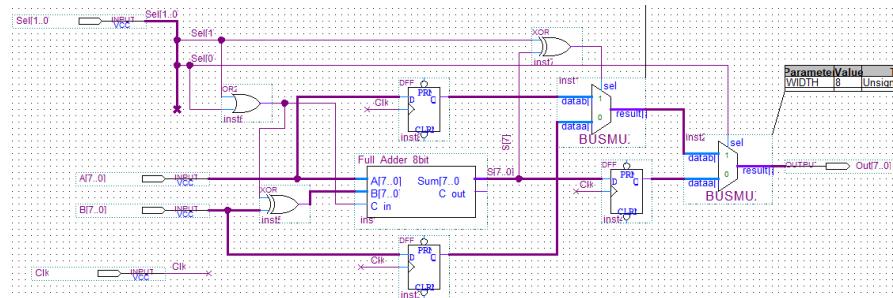


Figure 191: Task 7 AU2\_2Stage circuit

Operation table

Sel2	Sel1	Sel0	function
0	0	0	Add
0	0	1	
0	1	0	
0	1	1	
1	0	0	Abs(B)
1	0	1	Sub
1	1	0	Min
1	1	1	max

### 7.2.3. Full\_Adder\_8bit

Symbol

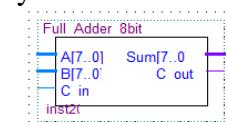


Figure 192: Task 7 Full\_Adder\_8bit symbol

Circuit

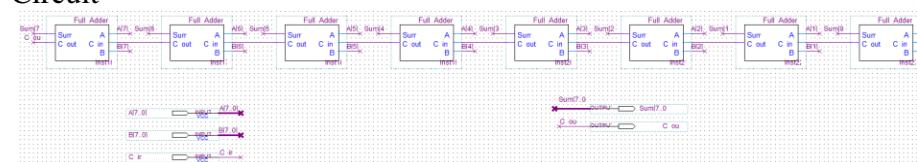
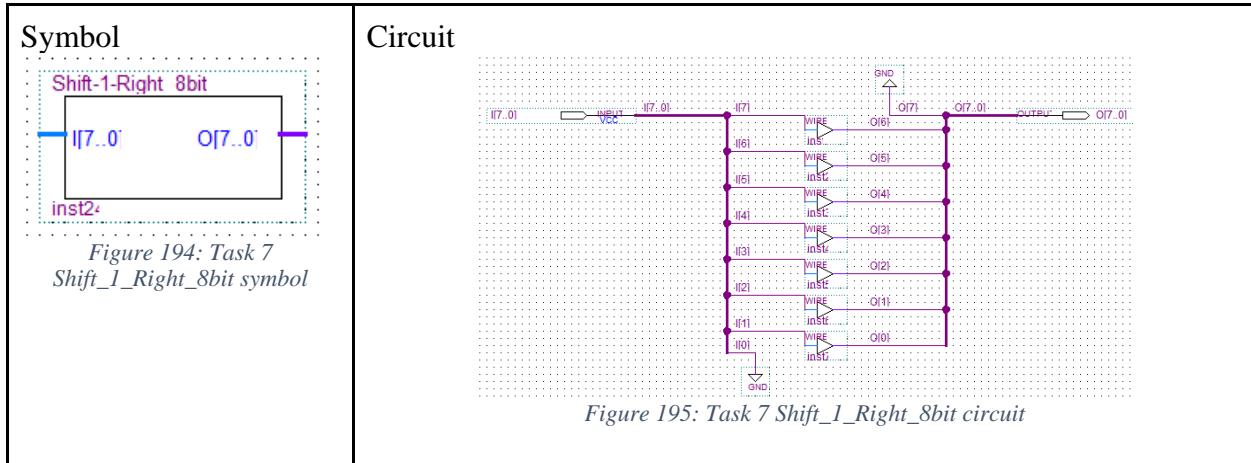
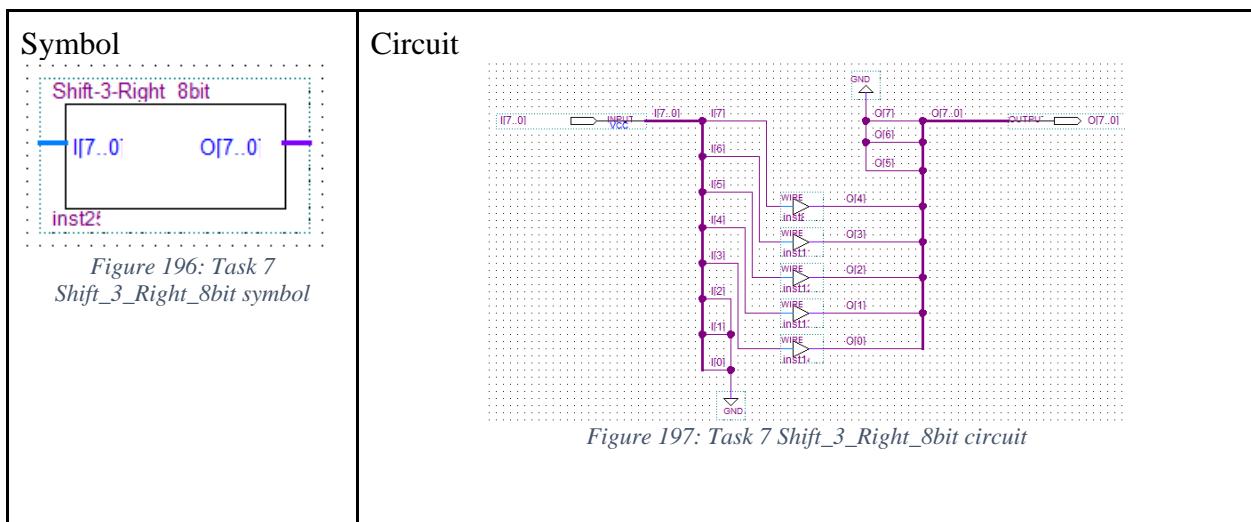


Figure 193: Task 7 Full\_Adder\_8bit circuit

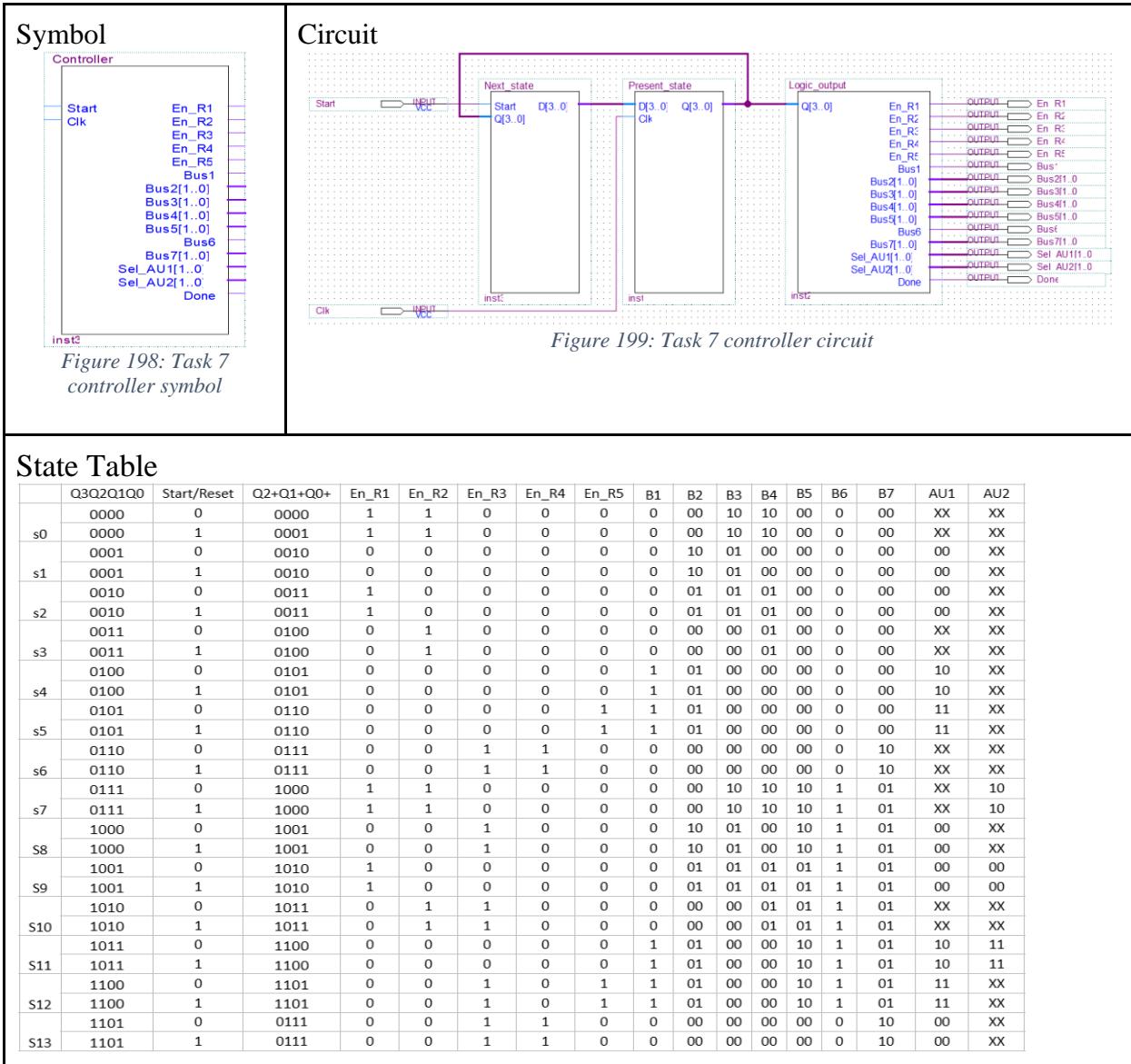
#### 7.2.4. Shift-1-Right\_8bit



#### 7.2.5. Shift-3-Right\_8bit



### 7.3. Controller Design



State Diagram

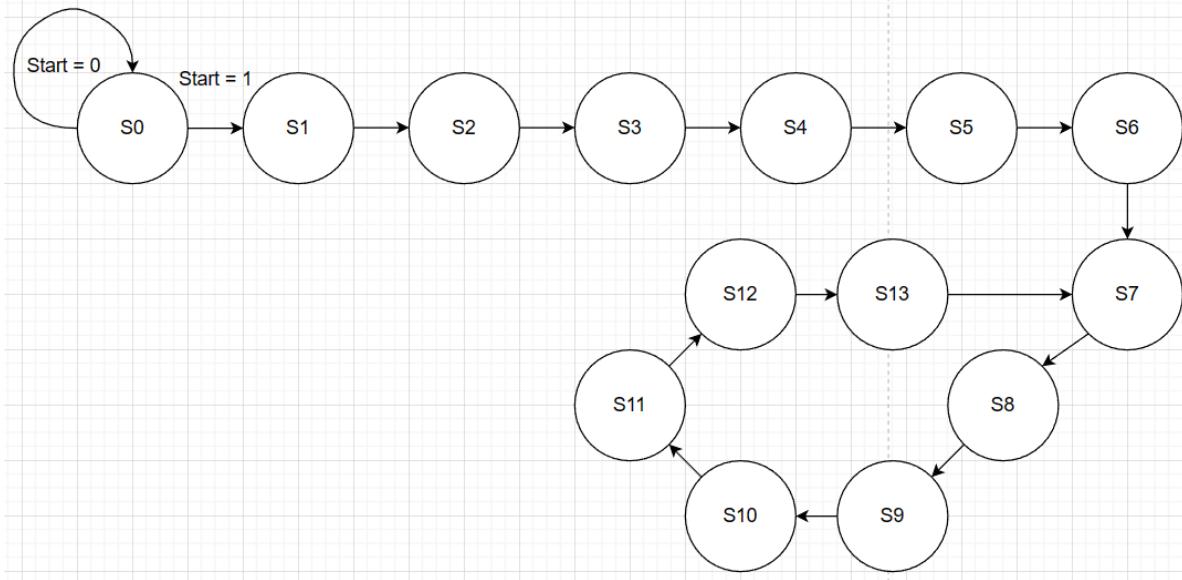
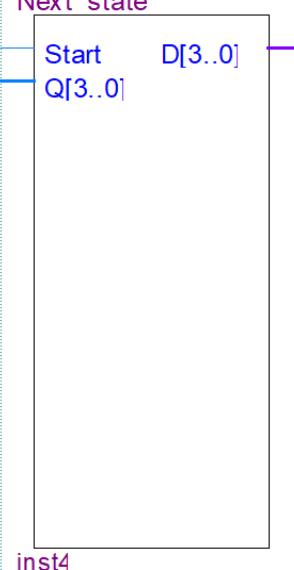
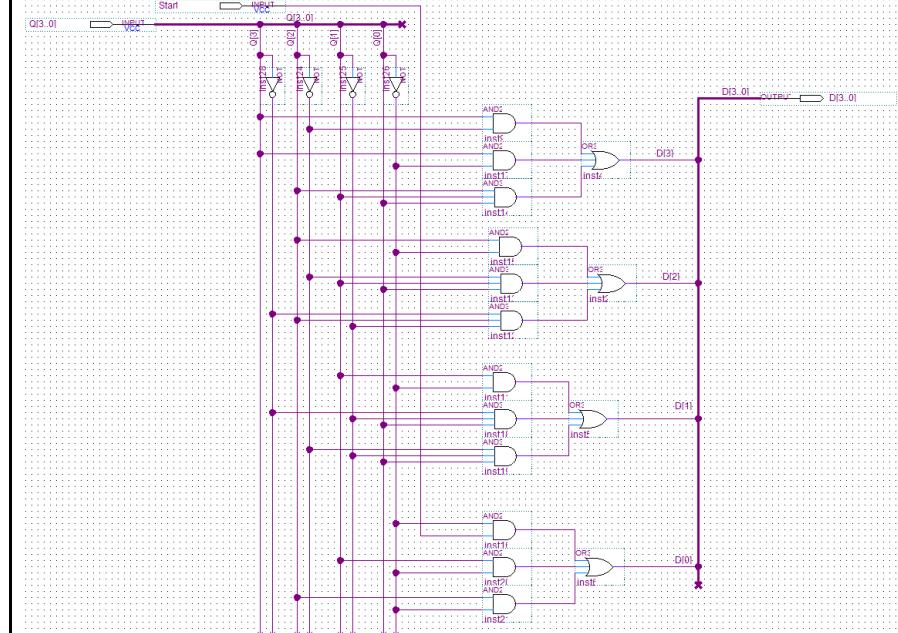


Figure 200: Task 7 state diagram

### 7.3.1. Next\_state

Symbol	Circuit
<b>Next state</b>  Start      D[3..0]  <i>inst4</i>	
<i>Figure 201: Task 7 Next_State symbol</i>	<i>Figure 202: Task 7 Next_State circuit</i>

Next\_state equation  

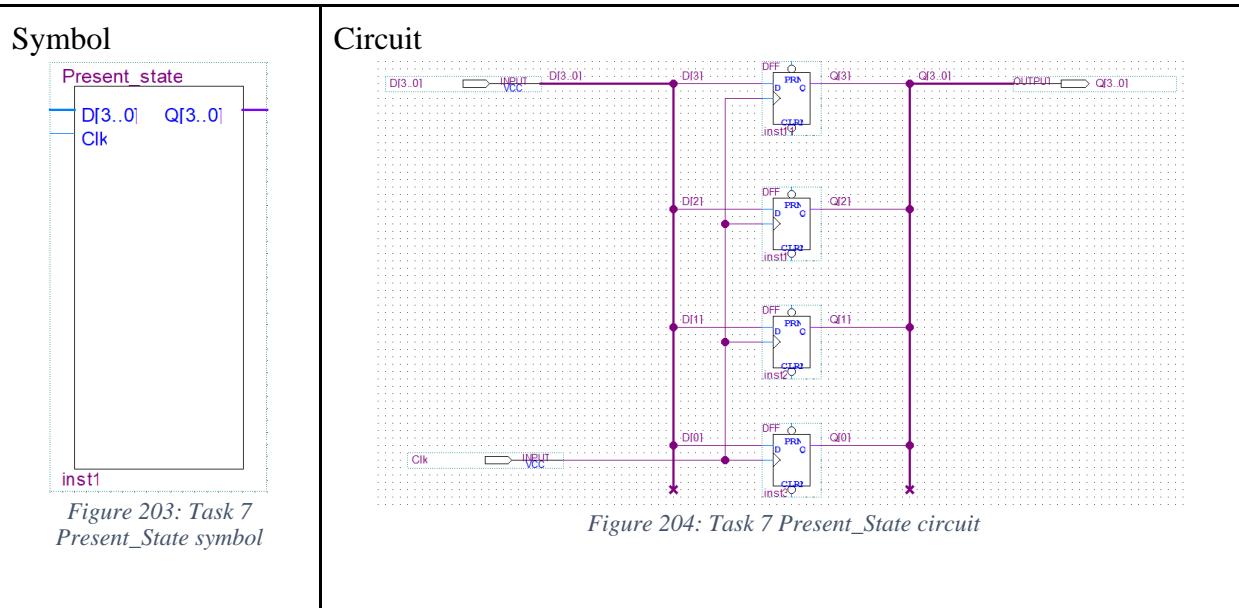
$$Q3+ = Q3.Q2' + Q3.Q0' + Q2.Q1.Q0$$

$$Q2+ = Q2.Q1' + Q2.Q0' + Q2'.Q1.Q0$$

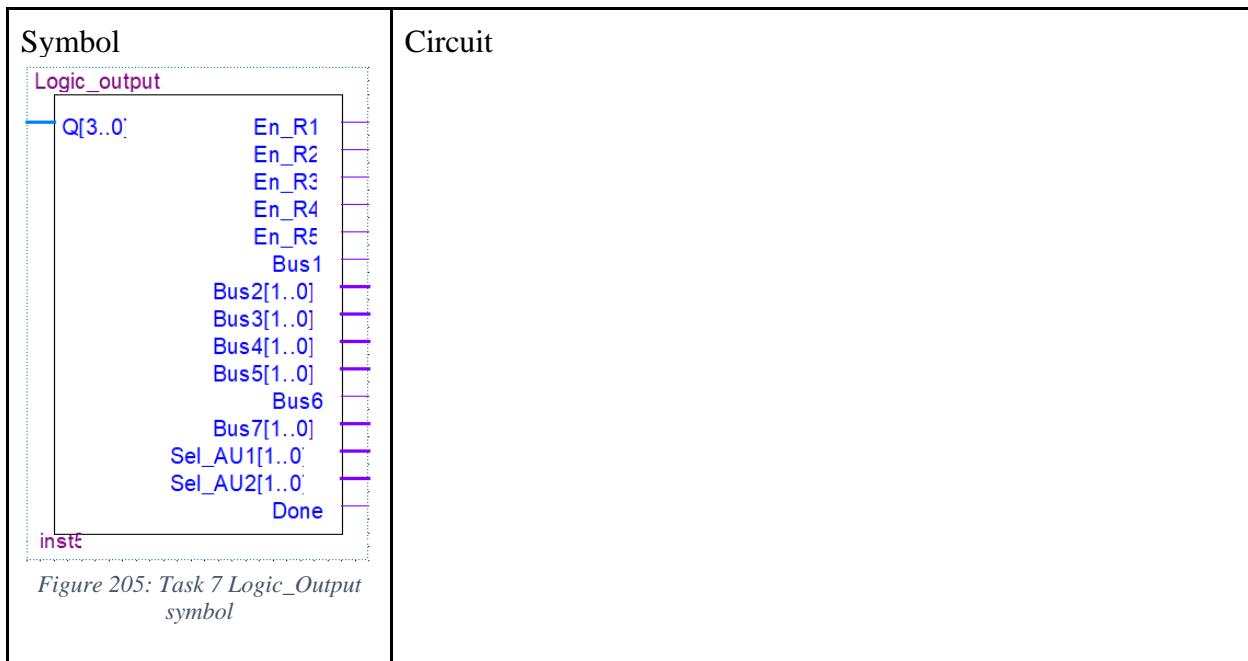
$$Q1+ = Q1'.Q0 + Q1.Q0'$$

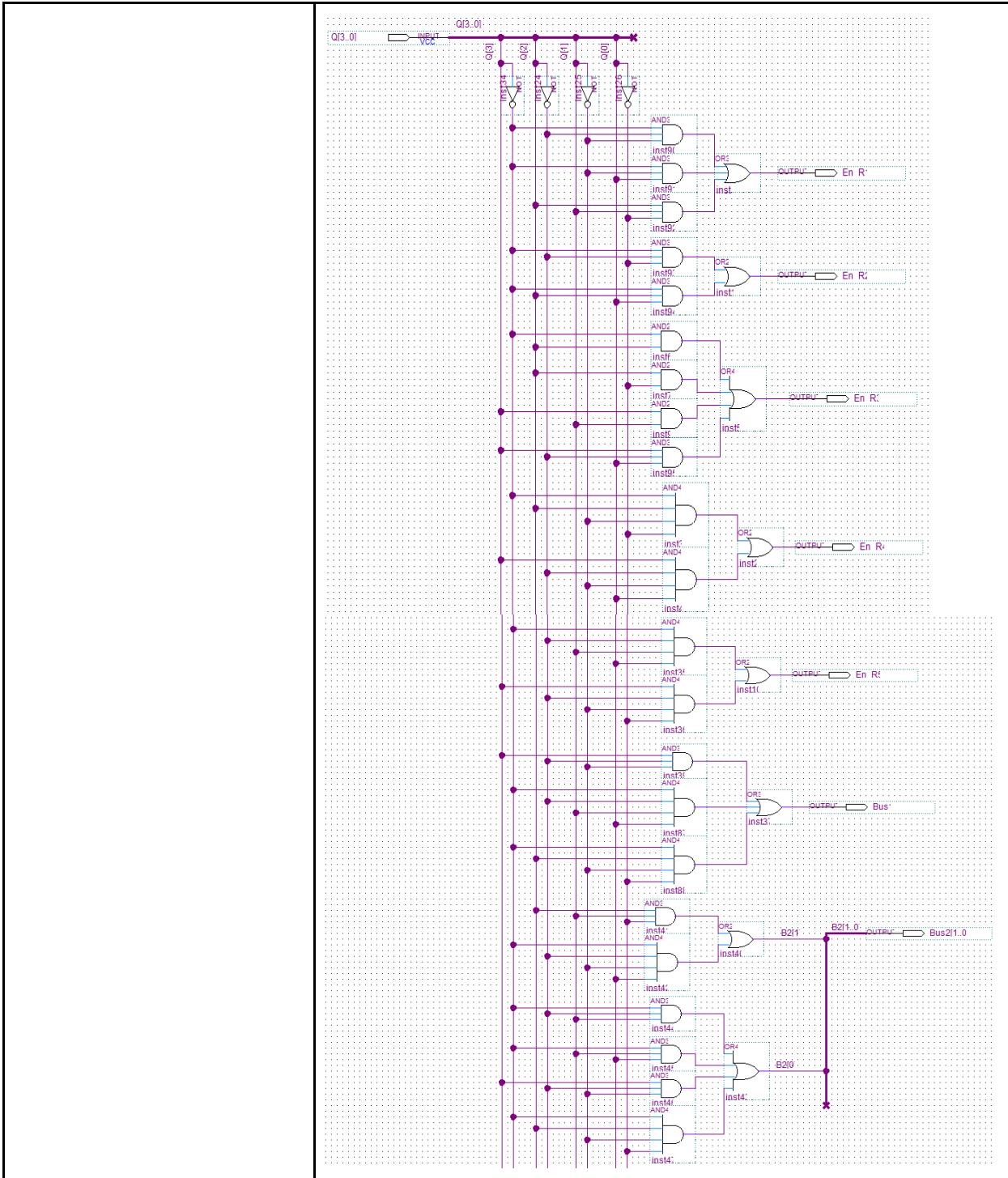
$$Q0+ = Q0'.Start + Q1.Q0' + Q2.Q0' + Q3.Q0' + Q3.Q2$$

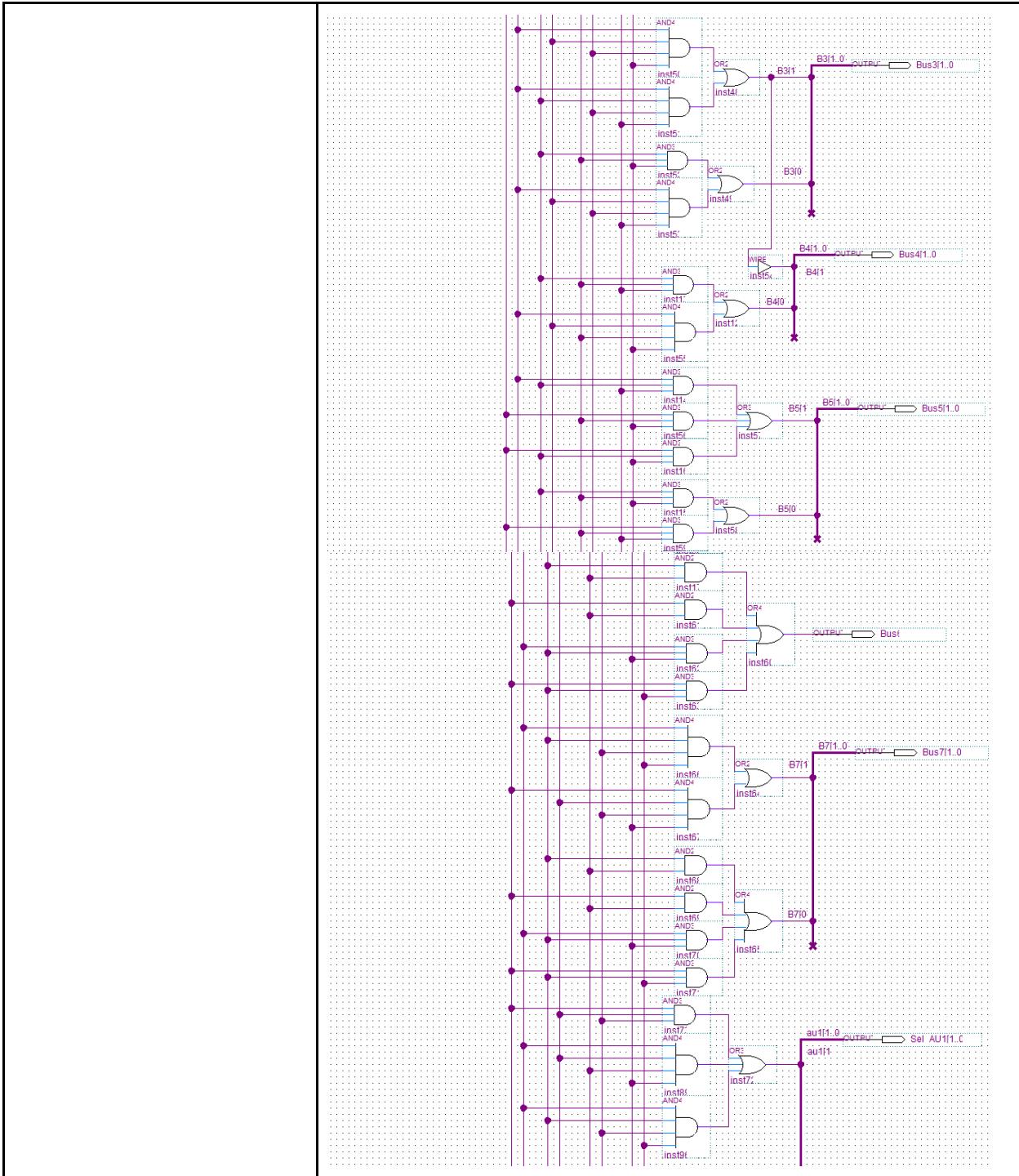
### 7.3.2. Present\_state



### 7.3.3. Logic\_output







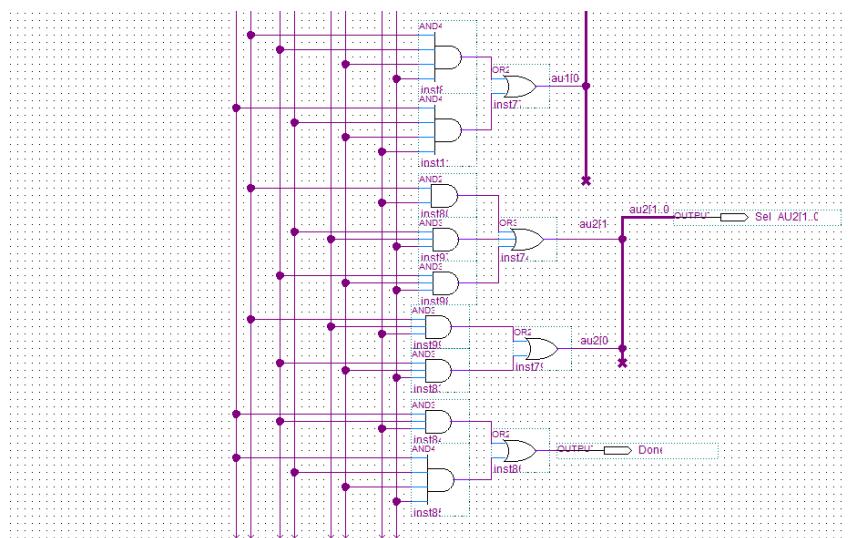


Figure 206: Task 7 Logic\_Output circuit

#### Logic\_output equation

$$En\_R1 = Q3'.Q2'.Q0' + Q2.Q1.Q0 + Q3.Q2'.Q1'.Q0$$

$$En\_R2 = Q3'.Q1.Q0 + Q3.Q1.Q0' + Q3'.Q2'.Q1'.Q0'$$

$$En\_R3 = Q3.Q0' + Q3.Q2 + Q2.Q1.Q0'$$

$$En\_R4 = Q2.Q1.Q0' + Q3.Q2.Q0$$

$$En\_R5 = Q3.Q2.Q0' + Q3'.Q2.Q1'.Q0$$

$$B1 = Q3'.Q2.Q1' + Q2.Q1'.Q0' + Q3.Q1.Q0$$

$$B2[1] = Q3'.Q2'.Q1'.Q0 + Q3.Q2'.Q1'.Q0'$$

$$B2[0] = Q3'.Q2.Q1' + Q2.Q1'.Q0' + Q3.Q2'.Q0 + Q3'.Q2'.Q1.Q0'$$

$$B3[1] = Q2.Q1.Q0 + Q3.Q2'.Q1'.Q0'$$

$$B3[0] = Q2'.Q1'.Q0 + Q3.Q2'.Q1' + Q3'.Q2'.Q1.Q0'$$

$$B4[1] = Q3'.Q2'.Q1'.Q0' + Q2.Q1.Q0$$

$$B4[0] = Q3'.Q2'.Q1 + Q2'.Q1.Q0' + Q3.Q2'.Q1'.Q0$$

$$B5[1] = Q3.Q1'.Q0' + Q2.Q1.Q0 + Q3.Q1.Q0$$

$$B5[0] = Q3.Q1.Q0' + Q3.Q2'.Q1'.Q0$$

$$B6 = Q3.Q2' + Q3.Q0' + Q2.Q1.Q0$$

$$B7[1] = Q2.Q1.Q0' + Q3.Q2.Q0$$

$$B7[0] = Q3.Q2' + Q3.Q0' + Q2.Q1.Q0$$

$$ALU1[1] = Q1.Q0 + Q3'.Q2 + Q2.Q0'$$

$$ALU1[0] = Q3'.Q2.Q0 + Q3.Q2.Q0'$$

$$ALU2[1] = Q1$$

$$ALU2[0] = Q2'.Q1$$

## 7.4. Waveform simulation

With 3 case: (-6, -8), (-3, 4) and (12, 16)

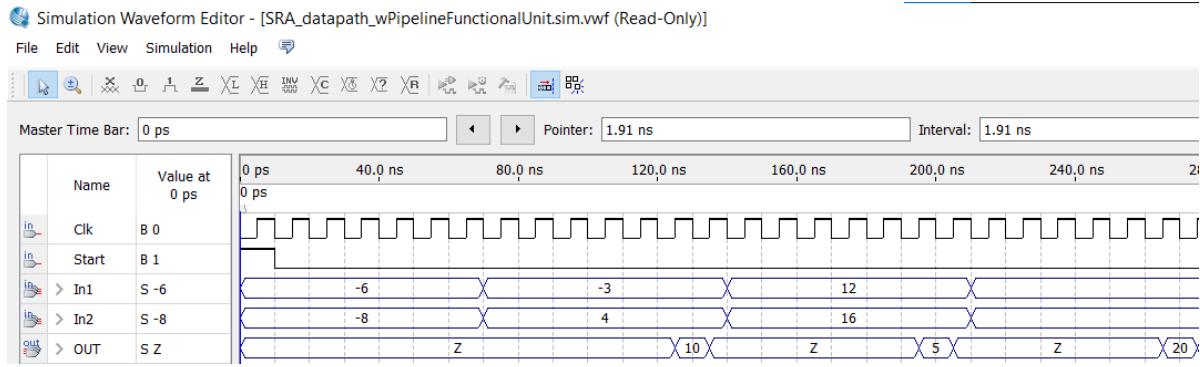


Figure 207: Task 7 waveform