Project 1

Create Your Own WIMP51 Version

INC A and INC Rn

Table of Contents

For this project, a version of the WIMP51 was created for the Altera DE2 FPGA board in Quartus II. This variation included two new instructions, increment of the accumulator and increment of a given register. In the following paragraphs, the changes to the original program and the reasons for the changes are documented below.

First, the increment of the accumulator was implemented. Using Appendix H of the Mazidi text, the INC A instruction was found to have the hex code of 04 and only required one byte. For this instruction, changes were made to the ALU. To start out, an 8:2:1 MUX was added to the ALU block diagr tell the program if it receives 0000 0100, t s put into the select line of the multiplexer. .

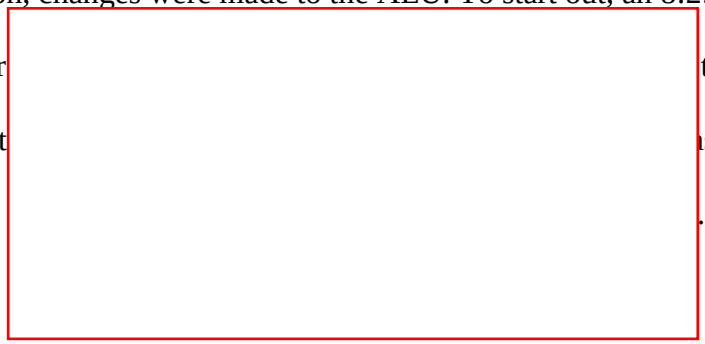


Figure 1: Desired Function for INC A

The output of the multiplexer was then fed back into the ripple-adder as before. This is shown below in Figure 2.

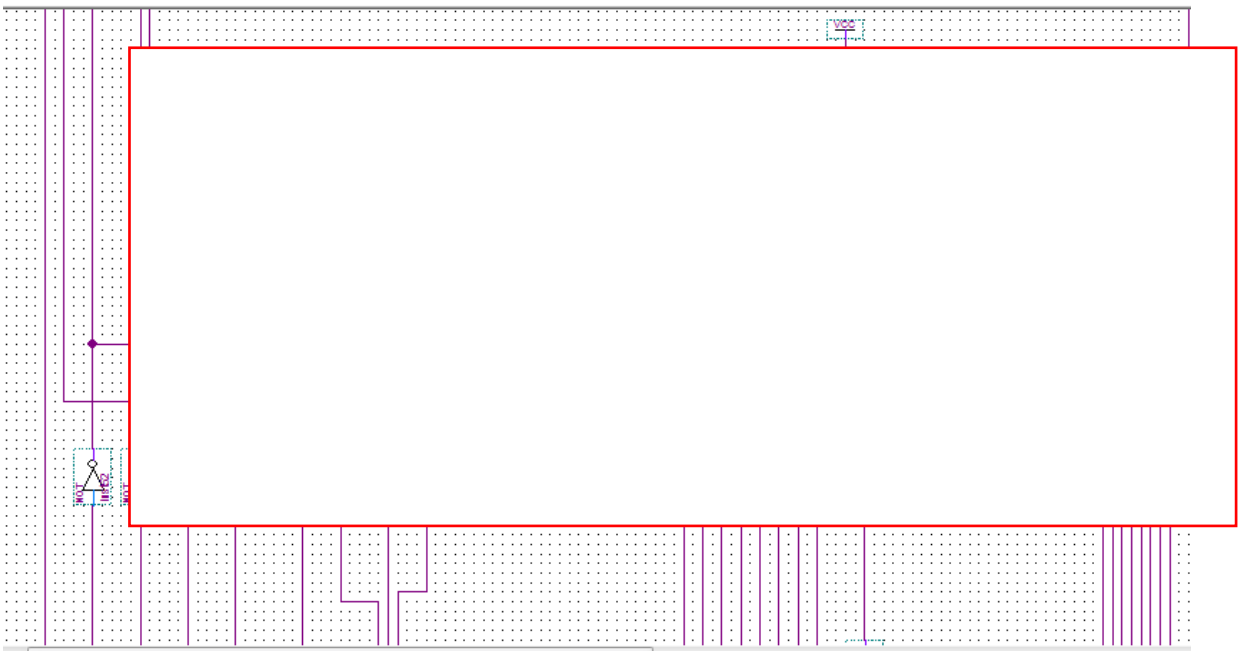


Figure 2: 8:2:1 MUX and AND Gate Added to ALU for INC A

Next, the output of the AUX Register was run into the 1 input section of the multiplexer while the 0 input section was 1 to the accumulator if it receiv ss the AUX straight through as if n LU can be found below in Figure 3.

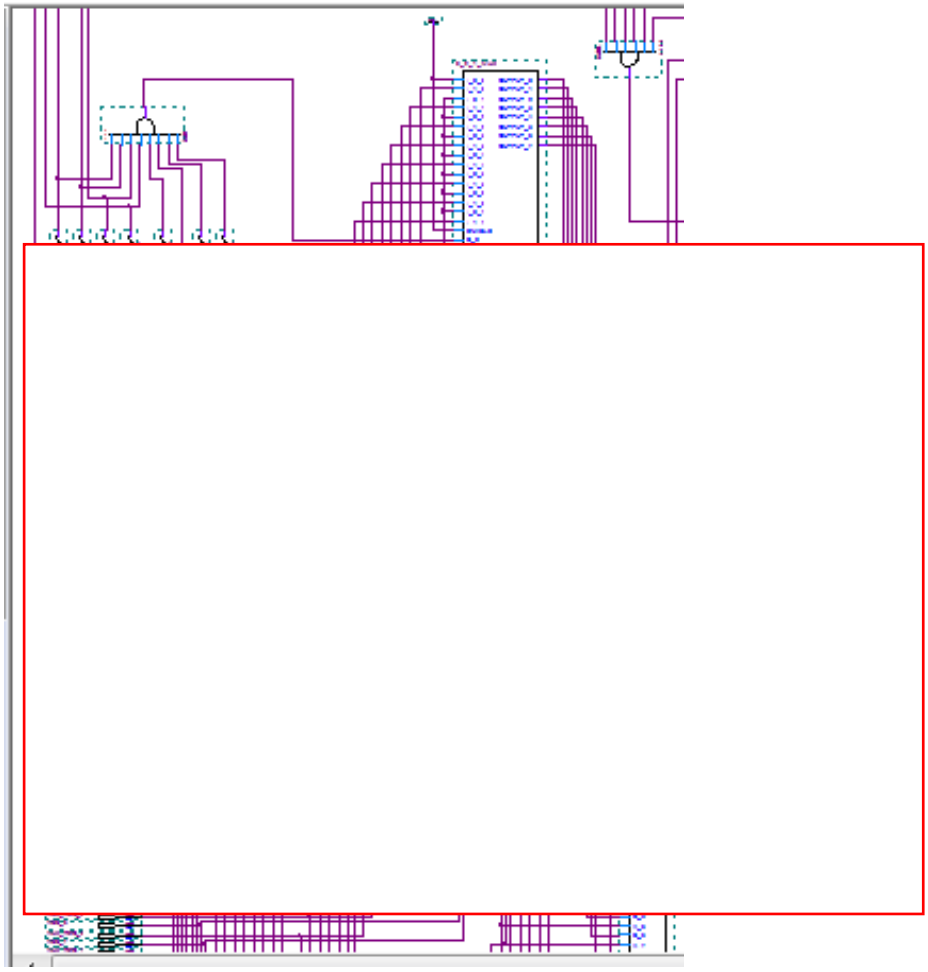


Figure 3: Overall Additions to ALU for INC A

Lastly, the L_A Select block diagram file was slightly modified by an OR Gate was added to output if it received any of the preloaded codes or if it got the new INC A instruction. This is shown below in Figure 4.

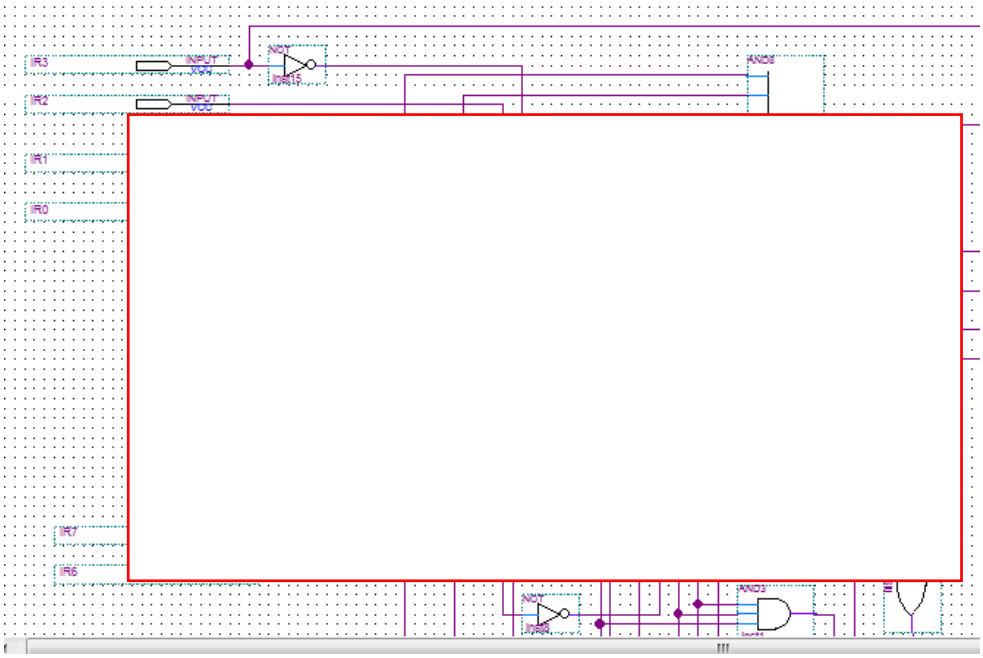
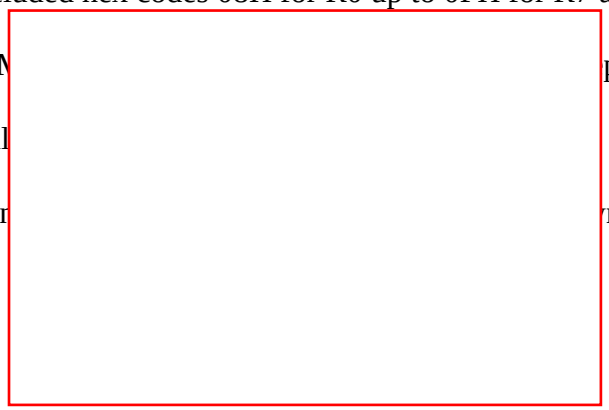


Figure 4: Changes Made to the L_A_SEL Block for INC A

For the next part of the project, INC Rn was implemented. After consulting the text, it was found that INC Rn included hex codes 08H for R0 up to 0FH for R7 and it was also a one byte instruction. An 8:2:1 M pt this time, the ACC output was fed into the mul line was fed off another AND gate consistir n below in Figure 5.



The output of the MUX was fed back into the ripple-adder, the same as with INC A. This can be found below in Figure 6.

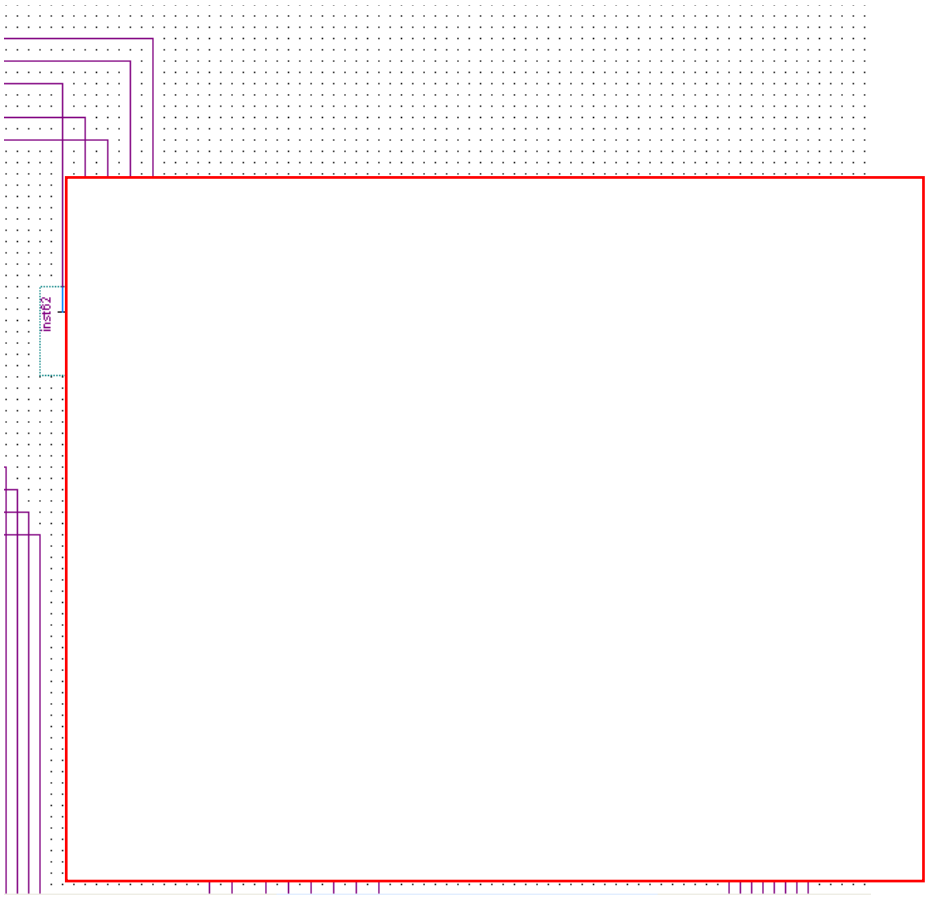


Figure 6: 8:2:1 MUX and AND Gate Added to ALU for INC Rn

The following figure shows the overall changes added to the ALU for the register increment.

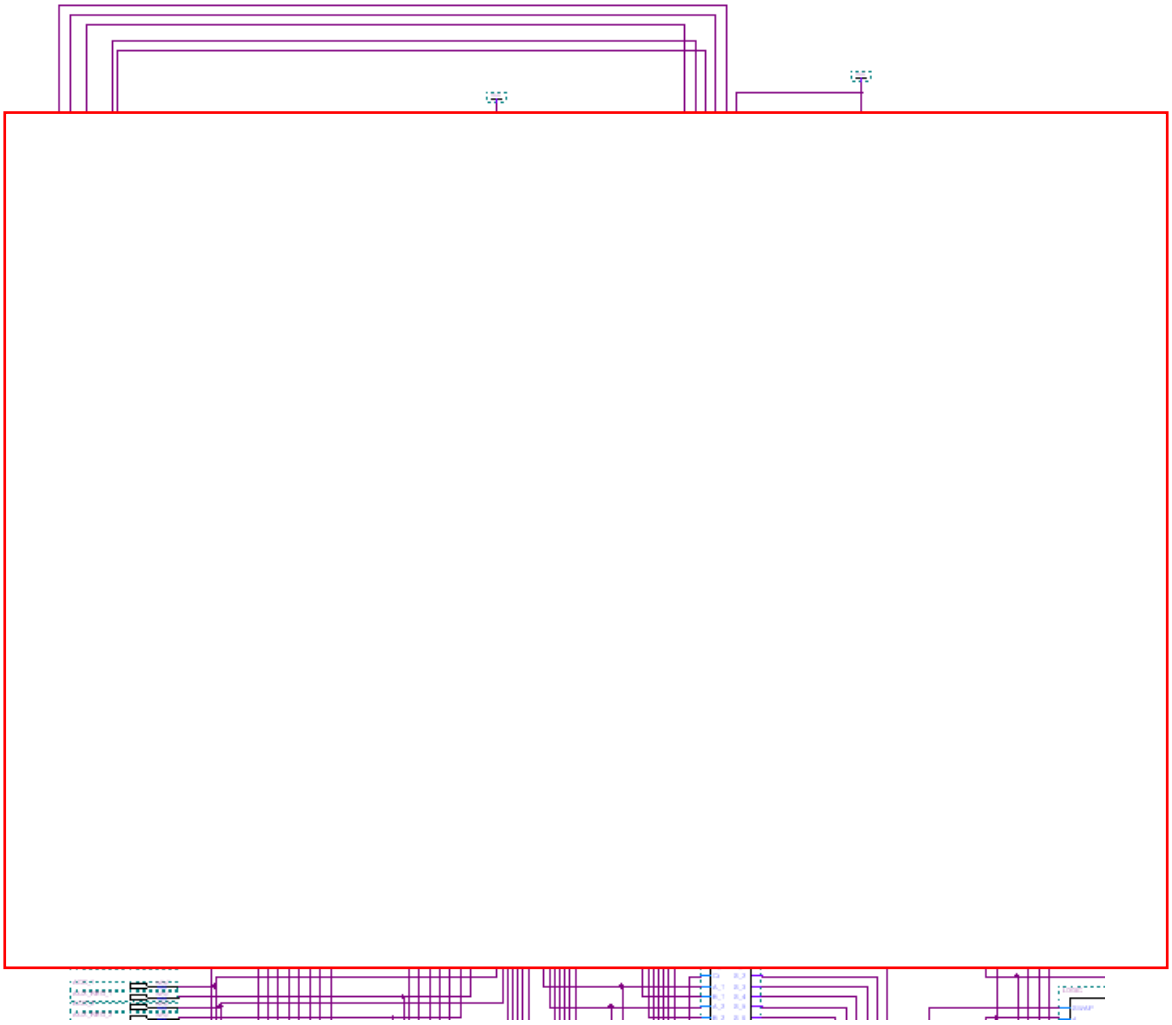


Figure 7: Overall Additions to ALU for INC Rn

The L_A Select block also had to be changed for this instruction so another branch was added to the OR gate to allow this code to pass through, as shown in Figure 8.

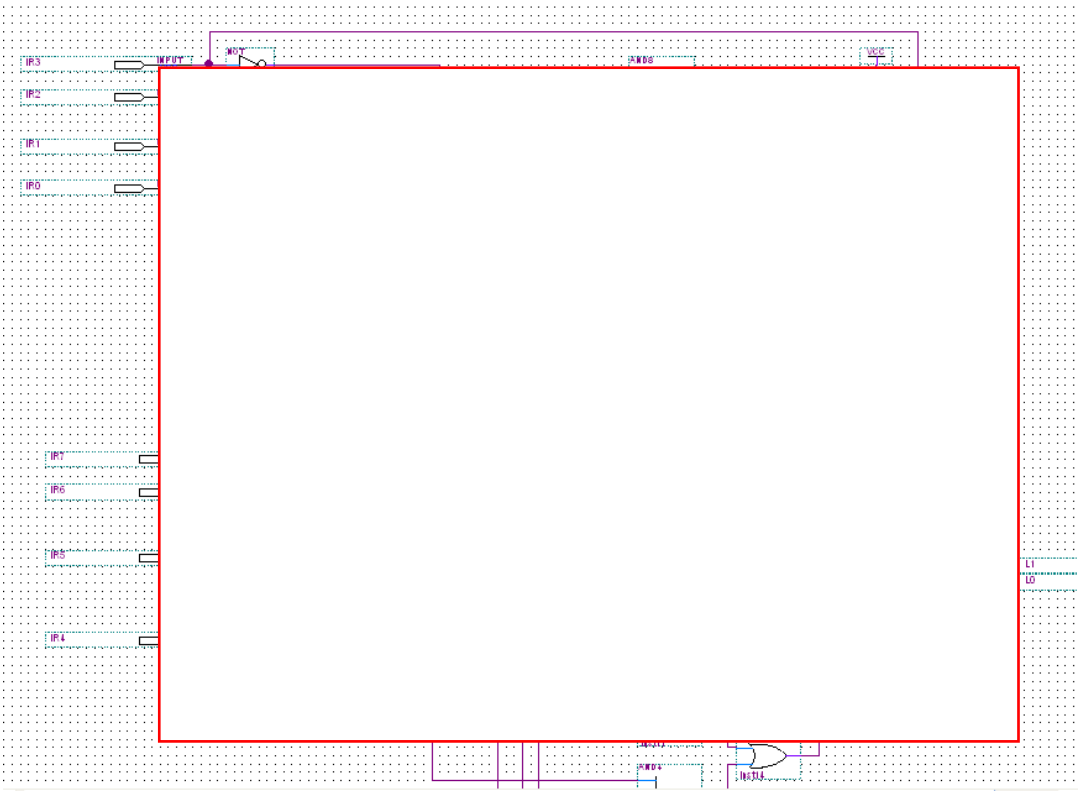


Figure 8: Changes Made to the L_A_SEL Block for INC Rn

Afterwards, the write-enable for the accumulator had to be modified for the program to implement properly by adding an extra branch off the existing NOR gate and incorporating the 0000 1xxx code. This is shown below in Figure 9.

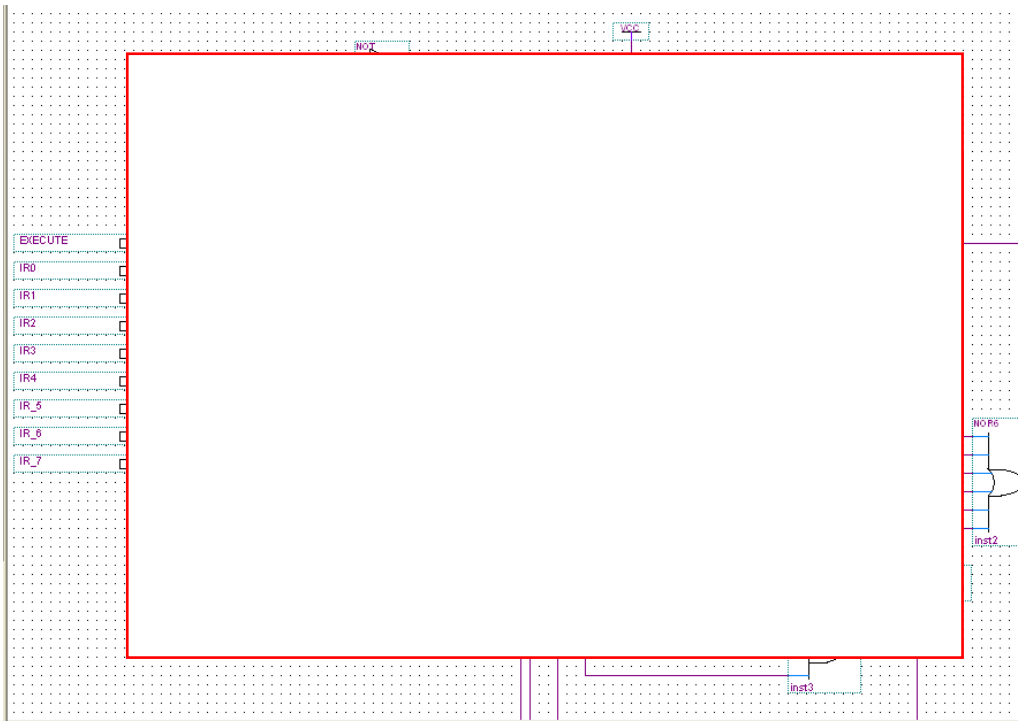


Figure 9: Changes Made to the ACC_WE Block for INC Rn

The register write enable file also had to be modified in order to incorporate the new instruction.

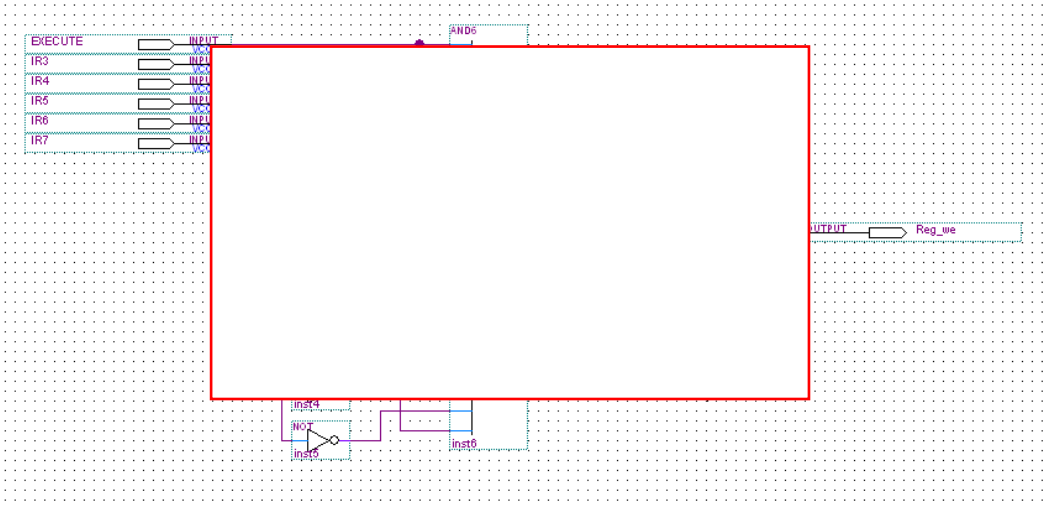


Figure 10: Changes Made to the REG_WE Block for INC Rn

Lastly, some changes had to be made on the overall Block2 file. Another 8:2:1 MUX had to be added in which the accumulator outputs of the ALU block were rerouted to one of the input locations of the multiplexer along with the outputs of the accumulator block. Once again, the

Select line was activated by the 0000 1xxx code and the output of the MUX was sent into the Top Register location. This is shown below in Figure 11.

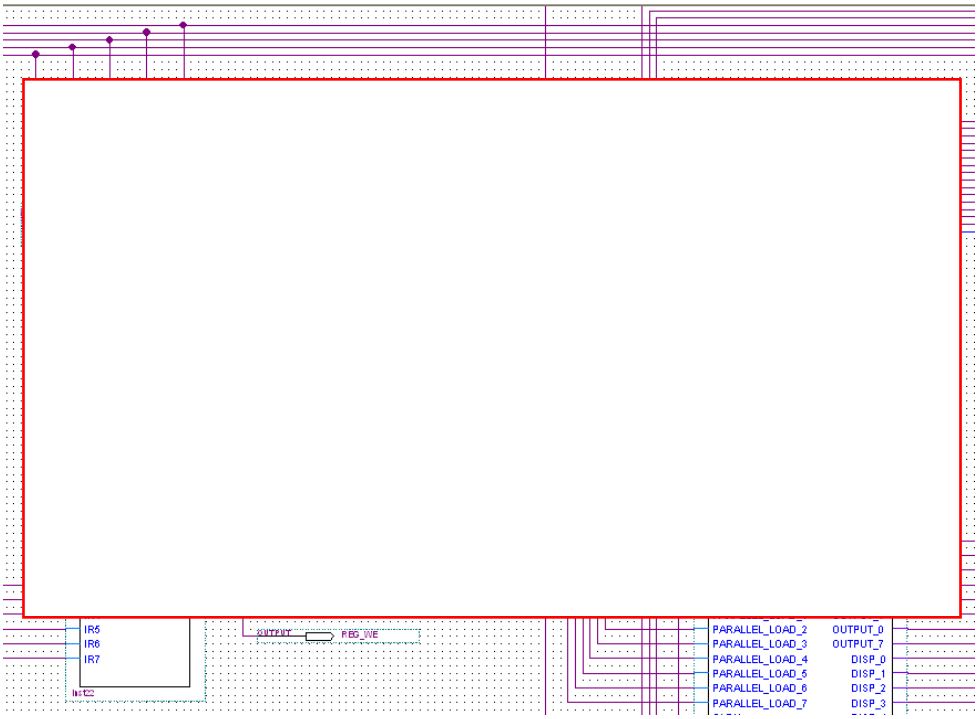


Figure 11: Additional 8:2:1 MUX Added to Block2 Used to Update the Register Values

After implementing these changes, INC A and INC Rn were successfully compiled and tested on the Altera board. The original instructions were also tested and no instructions seemed to be affected by the changes made. Included below in Appendix A is the program code used to test the variation program.

Appendix A: Test Program 1

00H	MOV A, #D	74H	
01H	04H	04H	A=04H
02H	CLR C	C3H	C=0
03H	SETB C	D3H	C=1
04H	ADDC A, #D	34H	
05H	03H	03H	A=08H
06H	INC A	04H	A=09H
07H	MOV R0, A	F8H	R0=09H
08H	SWAP A	C4H	A=90H
09H	MOV R1, A	F9H	R1=90H
0AH	INC R0	08H	R0=0AH
0BH	INC R1	09H	R1=91H
0CH	MOV A, R0	E8H	A=0AH
0DH	ORL A, R1	49H	A=9BH
0EH	ANL A, R0	58H	A=0AH
0FH	XRL A, R3	6BH	A=0AH
10H	CLR C	C3H	C=0
11H	ADDC A, R0	38H	A=14H
12H	SJUMP	80H	
13H	FEH	FEH	

Appendix A: Test Program 2

00H	INC R0	08H	R0=01H
01H	INC R1	09H	R1=01H
02H	INC R2	0AH	R2=01H
03H	INC R3	0BH	R3=01H
04H	INC R4	0CH	R4=01H
05H	INC R5	0DH	R5=01H
06H	INC R6	0EH	R6=01H
07H	INC R7	0FH	R7=09H
08H	SJUMP	80H	
09H	FEH	FEH	