Matthew Simiele
Nevin Nedumthakady
Project 1 Documentation
April 19th, 2018

The instruction format used for this project was a 14 bit code broken down as follows:

1. Rs (3:0)
2. Rd (7:4)
3. OPCODE (10:8)
4. Wenable (11)
5. Op Select (13:12)

Instruction to be Completed			
Wenable (11)	Opcode (10:8)	Mnemonic	Operation Performed
0	000	LD	Rd <= Rs
0	001	XOR	Rd <= Rd ^ Rs
0	010	AND	Rd <= Rd & Rs
0	011	OR	Rd <= Rd | Rs
0	100	ADD	Rd <= Rd + Rs
0	101	MULT	Rd <= Rd * Rs
0	110	INC	Rd <= Rd + 1
0	111	SUB	Rd <= Rd - Rs
1	000	JMP	PC <= Addr
1	001	BRCC	if C=0, PC <= Addr
1	010	BRCS	if C=1, PC <= Addr
1	011	BRNE	if Z=0, PC <= Addr
1	100	BREQ	if Z=1, PC <= Addr
1	101	BRPL	if N=0, PC <= Addr
1	110	BRMI	if N=1, PC <= Addr
1	111	CP	Rd <= Rd - Rs (does not write to Rd)

Input Select	
OP Select (13:12)	Input to the B Port of the DATA ALU
00	RS
01	(Input from Switches)
10	Input from Inch Counter
11	Constant 128

Description of the Instruction set used for the demo:

The program must be able to take inputs from the switches in order to be given a distance to travel so on the instruction set below PC 1-11 accomplishes this goal. The first loop asks for a user input that will only be accepted and the program will move on when the value is greater than 128, this is the magnitude of the Y component. The second loop determines the direction to travel, later in the program we will determine what way it will move based on the input. The loop will only exit when you flip the most significant or bit that represents 128 down. The next two loops are equivalent to the first two but will represent the x values instead of y.

The next part of the instruction sets the turn constant by doing math to R9, sets to 9.

After the turn constant is set an instruction clears the 8th bit of your magnitude register. When you x-or the value in your magnitude register with 128 you will clear the left most bit and leave all the other bits the same.

During the program an XOR R12,R12 will be used to clear the Inch Count after the inch count is cleared a check will be performed on a register that has a value of zero in it so you will not move on in the program until the inch count is not fully cleared.

The next step is to tell the rover to move forward or backward, the direction register holds a value between 127 and 0 when that value is checked with zero if the number is not zero the rover will move in the forward direction if it equals zero it will move backward.

After the first movement the program checks to see if there is a value in the other magnitude, if there is not the program jumps to the end. If there is a value the program moves on to the turn phase but not before the values in the x direction registers get moved to the old y direction registers and then after the values are moved the x direction registers are cleared so the rover does not turn again.

The rover performs a turn given the constant turn value computed in R9 and when the turn is completed it jumps back to the clear left most bit instruction

INSTRUCTION SET FOR DEMO:

ASSEMBLY			Machine Code					
PC	Label	Instruction	OpSel	W Enable	OPCODE	Rd	Rs	Hex
0	LD R1	LDI R1	01	0	000	0001	0000	1010
1		CP R1,128	11	1	111	0001	0000	3F10
2		BRCS LD R1	00	1	010	0000	0000	0A00
3	LD YD	LDI R4	01	0	000	0100	0000	1040
4		CP R4,128	11	1	111	0100	0000	3F40
5		BRCC LD YD	00	1	001	0000	0011	0903
6	LD R2	LDI R2	01	0	000	0010	0000	1020
7		CP R2,128	11	1	111	0010	0000	3F20
8		BRCS LD R2	00	1	010	0000	0110	0A06
9	LD XD	LDI R5	01	0	000	0101	0000	1050
10		CP R5,128	11	1	111	0101	0000	3F50
11		BRCC LD XD	00	1	001	0000	1001	0909
12	SET REG 9	INC R9	00	0	110	1001	0000	0690
13		ADD R9, R9	00	0	100	1001	1001	0499
14		ADD R9, R9	00	0	100	1001	1001	0499
15		ADD R9, R9	00	0	100	1001	1001	0499
16		INC R9	00	0	110	1001	0000	0690
17	Remove (RM) 8th Bit	XOR R1,128	11	0	001	0001	0000	3110
18	CLR IC0	XOR R12,R12	00	0	001	1100	1100	01CC
19		CPI R12, IC	10	1	111	1100	0000	2FC0
20		BRNE CLR IC0	00	1	011	0001	0010	0B12
21		CP R4,R12	00	1	111	0100	1100	0F4C
22		BRNE DF	00	1	011	0010	0000	0B20
23	BACK	AND R15,R15	00	0	010	1111	1111	02FF

24		CP R1, IC	10	1	111	0001	0000	2F10
25		BRCC BACK	00	1	001	0001	0111	0917
26	BREAK	AND R11,R11	00	0	010	1011	1011	02BB
27	CLR IC1	XOR R12,R12	00	0	001	1110	1110	01CC
28		CPI R12, IC	10	1	111	1100	0000	2FC0
29		BRNE CLR IC1	00	1	011	0001	1011	0B1B
30		JMP TURN	00	1	000	0010	0111	0827
31		MOV R0,R0	00	0	000	0000	0000	0000
32	Drive Forward (DF)	AND R14,R14	00	0	010	1110	1110	02EE
33		CP R1,IC	10	1	111	0001	0000	2F10
34		BRCC FOR	00	1	001	0010	0000	0920
35	BREAK	AND R11,R11	00	0	010	1011	1011	02BB
36	CLR IC2	XOR R12,R12	00	0	001	1100	1100	01CC
37		CPI R12, IC	10	1	111	1100	0000	2FC0
38		BRNE CLR IC2	00	1	011	0010	0100	0B24
39		AND R2,R2	00	0	010	0010	0010	0222
40		BREQ END	00	1	100	0011	0010	0C32
41		MOV R1,R2	00	0	000	0001	0010	0012
42		XOR R2,R2	00	0	001	0010	0010	0122
43		MOV R4,R5	00	0	000	0100	0101	0045
44		XOR R5,R5	00	0	001	0101	0101	0155
45	TURN	AND R13,R13	00	0	010	1101	1101	02DD
46		CPI R9,IC	10	1	111	1001	0000	2F90
47		BRCC TURN	00	1	001	0010	1101	092D
48	BREAK	AND R11,R11	00	0	010	1011	1011	02BB

49		JMP RM 8th Bit	00	1	000	0001	0001	0811
50	END	CLR R9	00	0	001	1001	1001	0199