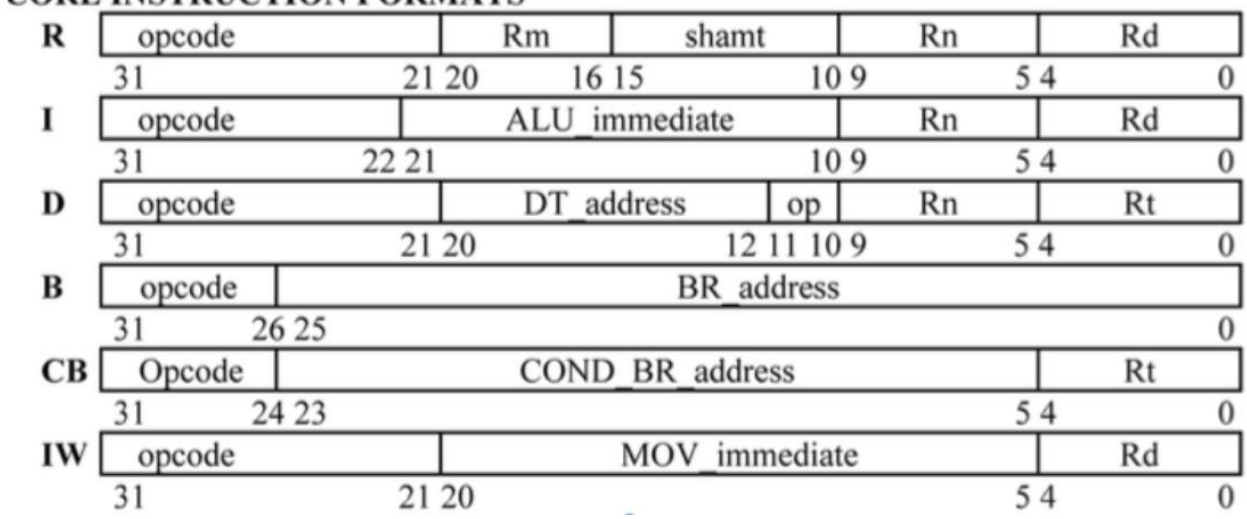
LEGv8 Test Program for CDA-4101: Project 2

byte idx	word idx	Assembly Instruction	Binary Instruction 31 ..... 0 10987654321098765432109876543210	Hex code B3B2B1B0	Description
0	0	main: ADD X0, XZR, XZR	10001011000 11111 000000 11111 00000	8b1f03e0	X0 <= 0 = DM.startAdds
4	1	LDUR X2, [X0,#0]	11111000010 000000000 00 00000 00010	f8400002	X2 <= 5 = DM[0+0]
8	2	LDUR X3, [X0,#8]	11111000010 000001000 00 00000 00011	f8408003	X3 <= 12 = DM[0+8]
12	3	LDUR X7, [X0,#16]	11111000010 000010000 00 00000 00111	f8410007	X7 <= 3 = DM[0+16]
16	4	ORR X4, X7, X2	10101010000 00010 000000 00111 00100	aa0200e4	X4 <= 7 <= 3 OR 5
20	5	AND X5, X3, X4	10001010000 00100 000000 00011 00101	8a040065	X5 <= 4 <= 12 AND 7
24	6	ADD X5, X5, X4	10001011000 00100 000000 00101 00101	8b0400a5	X5 <= 11 <= 4 + 7
28	7	SUB X8, X5, X7	11001011000 00111 000000 00101 01000	cb0700a8	X8 <= 8 <= 11 - 3
32	8	CBZ X8, quit	10110100 0000000000000001011 01000	b4000168	should not branch
36	9	SUB X9, X3, X4	11001011000 00100 000000 00011 01001	cb040069	X9 <= 5 <= 12 - 7
40	10	CBNZ X9, next	10110101 0000000000000000010 01001	b5000049	should branch
44	11	ADDI X5, X0, 4	1001000100 000000000100 00000 00101	91001009	should not execute
48	12	next: SUB X8, X7, X2	11001011000 00010 000000 00111 01000	cb0200e8	X8 <= -2 <= 3 - 5
52	13	ADD X7, X4, X5	10001011000 00101 000000 00100 00111	8b050087	X7 <= 18 <= 7 + 11
56	14	SUB X7, X7, X2	11001011000 00010 000000 00111 00111	cb0200e7	X7 <= 13 <= 18 - 5
60	15	STUR X7, [X3,#52]	11111000000 000110100 00 00110 00111	f8034067	DM[12+52] <= 13
64	16	LDUR X2, [X0,#64]	11111000010 001000000 00 00000 00010	f8440002	X2 <= 13 <= DM[0+64]
68	17	B quit	000101 00000000000000000000000010	14000002	should branch
72	18	ADDI X2, X0, 6	1001000100 000000000110 00000 00010	91001802	should not execute
76	19	quit: STUR X2, [X0,#80]	11111000000 000101000 00 00000 00010	f8050002	DM[0+80] <= 13
80	20	HALT	11111111111 0000000000000000000000	ffe00000	stop execution

CORE INSTRUCTION FORMATS



List of opcodes to be supported

```
LDUR   11111000010 (11)

STUR   11111000000 (11)

ADD    10001011000 (11)

ADDI   1001000100  (10)

SUB    11001011000 (11)

AND    10001010000 (11)

ORR    10101010000 (11)

CBZ    10110100    (8)

CBNZ   10110101    (8)

B      000101      (6)

HALT   11111111111 (11)
```

Instruction Memory	
byte adds	**Byte value**
0	e0
1	03
2	1f
3	8b
4	02
5	00
6	40
7	f8
8	03
9	80
10	40
11	f8
12	07
13	00
14	41
15	f8
16	e4
17	00

Initial Data Memory Value	
byte adds	**Byte value**
0	05
1	00
2	00
3	00
4	00
5	00
6	00
7	00
8	0c
9	00
10	00
11	00
12	00
13	00
14	00
15	00
16	03
17	00

18	02
19	aa
20	65
21	00
22	04
23	8a
24	a5
25	00
26	04
27	8b
28	a8
29	00
30	07
31	cb
32	68
33	01
34	00
35	b4
36	69
37	00
38	04
39	cb
40	49
41	00
42	00
43	b5
44	09
45	10
46	00
47	91
48	e8
49	00
50	02
51	cb
52	87
53	00
54	05
55	8b
56	e7
57	00
58	02
59	cb
60	67
61	40
62	03

18	00
19	00
20	00
21	00
22	00
23	00
24	00
25	00
26	00
27	00
28	00
29	00
30	00
31	00
32	00
33	00
34	00
35	00
36	00
37	00
38	00
39	00

63	f8
64	02
65	00
66	44
67	f8
68	02
69	00
70	00
71	14
72	02
73	18
74	00
75	91
76	02
77	00
78	05
79	f8
80	00
81	00
82	e0
83	ff