PC initially at 0

DRAM content at addresses 0 – 2, and 17 and 18

0	LODD 17	0000000000010001
1	ADDD 18	0010000000010010
2	PUSH	1111010000000000

17	63	0000000000111111
18	10	0000000000001010

```
0:mar := pc; rd;
1:pc := 1 + pc; rd;
2:ir := mbr; if n then goto 28;
      ir → 0000000000010001   NO JUMP
3:tir := lshift(ir + ir); if n then goto 19;
      check ir+ir                NO JUMP
      tir → 0000000001000100
4:tir := lshift(tir); if n then goto 11;
      check tir                  NO JUMP
      tir → 0000000010001000
5:alu := tir; if n then goto 9;
      check tir                  NO JUMP
      we must have a LODD instruction so
6:mar := ir; rd;
      mar → 000000010001   (mar is a 12 bit register)
7:rd;
8:ac := mbr; goto 0;
      ac → 0000000000111111   (value is 63 base 10)
```

PC now at 1

DRAM content at addresses 0 – 2, and 17 and 18

0	LODD 17	0000000000010001
1	ADDD 18	0010000000010010
2	PUSH	1111010000000000

17	63	0000000000111111
18	10	0000000000001010

```
0:mar := pc; rd;
1:pc := 1 + pc; rd;
2:ir := mbr; if n then goto 28;
      ir ➔ 0010000000010010  NO JUMP
3:tir := lshift(ir + ir); if n then goto 19;
      check ir+ir               NO JUMP
      tir ➔ 1000000001000100
4:tir := lshift(tir); if n then goto 11;
      check tir                 JUMP TO 11
      tir ➔ 0000000010001000
11:alu := tir; if n then goto 15;
      tir ➔ 0000000010001000  NO JUMP
      we must have a ADDD instruction so
12:mar := ir; rd;
      mar ➔ 000000010010   (value is 18)
13:rd;
14:ac := ac + mbr; goto 0;
      ac ➔  0000000000111111 + 0000000000001010 (63+10)
      ac ➔  0000000001001001                    (73)
```

PC now at 2

```
0   LODD 17     0000000000010001
1   ADDD 18     0010000000010010
2   PUSH        1111010000000000
```

```
0:mar := pc; rd;
1:pc := 1 + pc; rd;
2:ir := mbr; if n then goto 28;
      ir → 1111010000000000  JUMP TO 28
28:tir := lshift(ir + ir); if n then goto 40;
      check ir+ir                JUMP TO 40
      tir → 1101000000000000
40:tir := lshift(tir); if n then goto 46;
      check tir                  JUMP TO 46
      tir → 1010000000000000
46:tir := lshift(tir); if n then goto 50;
      tir → 0100000000000000  JUMP TO 50
50:tir := lshift(tir); if n then goto 65;
      tir → 1000000000000000  NO JUMP
51:tir := lshift(tir); if n then goto 59;
      tir → 0000000000000000  JUMP TO 59
59:alu := tir; if n then goto 62;
      tir → 0000000000000000  NO JUMP
60:sp := sp + (-1);
61:mar := sp; mbr := ac; wr; goto 10;
10:wr; goto 0;
```