The table below labeled 'A' represents a state for the processor. We show the machine instructions in decoded psuedocode form for simplicity and so we don't have to describe or reason about the processor architecture directly. They are intended to be single-cycle instructions on a simple processor like the one developed last week.

A

State		Value
PC		0
DMEM	0	5
	1	35
	2	255
	3	66
IMEM	0	DMEM[2]=DMEM[2]-DMEM[2]; PC=PC+1
	1	DMEM[1]=DMEM[2]+1; PC=PC+1
	2	DMEM[2]=DMEM[2]+DMEM[1]; PC=PC+1
	3	DMEM[1]=DMEM[1]+1; PC=PC+1
	4	DMEM[3]=DMEM[1]-DMEM[0]; PC=PC+1
	5	if (DMEM[3]!=0) PC=2 else PC=PC+1
	6	DMEM[0]=DMEM[2]; PC=PC+1
	7	RETURN

Starting with the processor in State A shown, what is the state for each of the next 12 cycles of operation. (No instructions change instruction memory, so we omit it here.)

First instruction execution shown.

		DMEM			
Cycle	**PC**	0	1	2	3
Initial	0	5	35	255	66
+1	1	5	35	0	66
+2					
+3					
+4					
+5					
+6					
+7					
+8					
+9					
+10					
+11					
+12					

We will now also look at a second task:

B

IMEM	0	DMEM[1]=DMEM[1]-DMEM[1]; PC=PC+1
	1	DMEM[1]=DMEM[1]-1; PC=PC+1
	2	if (DMEM[0]==0) PC=6 else PC=PC+1
	3	DMEM[1]=DMEM[1]+1; PC=PC+1
	4	DMEM[0]=DMEM[0]>>1 *(integer divide by 2; if get half, round down)*; PC=PC+1
	5	PC=2
	6	DMEM[0]=DMEM[1]; PC=PC+1
	7	RETURN

		IMEM	**DMEM**			
Cycle	**PC**		0	1	2	3
0	0	**B**	23	255	222	192
+1		**B**				
+2		**B**				
+3		**B**				
+4		**B**				
+5		**B**				
+6		**B**				
Swap in A+6 state (from previous side) in next line and continue						
+6		**A**				
+7		**A**				
+8		**A**				
+9		**A**				
+10		**A**				
+11		**A**				
+12		**A**				

A

IMEM	0	DMEM[2]=DMEM[2]-DMEM[2]; PC=PC+1
	1	DMEM[1]=DMEM[2]+1; PC=PC+1
	2	DMEM[2]=DMEM[2]+DMEM[1]; PC=PC+1
	3	DMEM[1]=DMEM[1]+1; PC=PC+1
	4	DMEM[3]=DMEM[1]-DMEM[0]; PC=PC+1
	5	if (DMEM[3]!=0) PC=2 else PC=PC+1
	6	DMEM[0]=DMEM[2]; PC=PC+1
	7	RETURN

B

IMEM		
	0	DMEM[1]=DMEM[1]-DMEM[1]; PC=PC+1
	1	DMEM[1]=DMEM[1]-1; PC=PC+1
	2	if (DMEM[0]==0) PC=6 else PC=PC+1
	3	DMEM[1]=DMEM[1]+1; PC=PC+1
	4	DMEM[0]=DMEM[0]>>1 *(integer divide by 2; if get half, round down)*; PC=PC+1
	5	PC=2
	6	DMEM[0]=DMEM[1]; PC=PC+1
	7	RETURN

		IMEM	**DMEM**			
Cycle	**PC**		0	1	2	3
Swap in B+6 state (from previous page) in next line and continue						
+6		**B**				
+7		**B**				
+8		**B**				
+9		**B**				
+10		**B**				
+11		**B**				
+12		**B**				
Swap in A+12 state (from previous page) in next line and continue						
+12		**A**				
+13		**A**				
+14		**A**				
+15		**A**				
+16		**A**				
+17		**A**				
+18		**A**				

A

IMEM		
	0	DMEM[2]=DMEM[2]-DMEM[2]; PC=PC+1
	1	DMEM[1]=DMEM[2]+1; PC=PC+1
	2	DMEM[2]=DMEM[2]+DMEM[1]; PC=PC+1
	3	DMEM[1]=DMEM[1]+1; PC=PC+1
	4	DMEM[3]=DMEM[1]-DMEM[0]; PC=PC+1
	5	if (DMEM[3]!=0) PC=2 else PC=PC+1
	6	DMEM[0]=DMEM[2]; PC=PC+1
	7	RETURN