Architecture Operating Systems Coursework 2

Adder/Subtractor Machine

Aimee Louise Lewis, Edward Gray, Riley Mangan and Louis Oregan

Step 1 : Cycle 1

CPU Registers

The 1st place in the register is for the opcode, an the latter 3 digits is for the Memory address that instruction uses.

PC	0	3	0	0
MAR				
MBR				
IR				
AC				

Opcodes

Load from Memory	1
Store to Memory	2
Add to Memory	3
Subtract to Memory	4

Memory

300				
301				
302				
303	.	.	.	.
304				
305	**0**	0	0	5
306	**0**	0	0	1

-305 and **306** contain the two numbers we will be **adding** in this example.
-Please not that the memory stores values in Nibbles, integers are for readability only.

Step 2

CPU Registers

The MAR has #300 in it now because the PC was at #300.

PC	0	3	0	0
MAR	0	3	0	0
MBR				
IR				
AC				

Address Bus

Opcodes

Load from Memory	1
Store to Memory	2
Add from Memory	3
Subtract from Memory	4

Memory

300	**1**	3	0	4
301	**3**	3	0	5
302	**2**	3	0	6
303	**.**	.	.	.
304	**0**	0	0	5
305	**0**	0	0	1
306				

I have now filled in all of the instructions that we will be performing as well as the memory from last slide.

Step 3

CPU Registers

The MAR points to #300 in memory - that data is now placed into the MBR.

PC	0	3	0	0
MAR	0	3	0	0
MBR	1	3	0	4
IR				
AC				

Data Bus

Opcodes

Load from Memory	1
Store to Memory	2
Add from Memory	3
Subtract from Memory	4

Memory

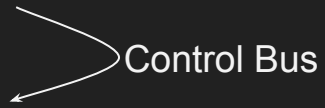
300	**1**	3	0	4
301	**3**	3	0	5
302	**2**	3	0	6
303	**.**	.	.	.
304	**0**	0	0	5
305	**0**	0	0	1
306				

Step 4

CPU Registers

Because the 1st value of the MBR is an instruction, the MBR is now loaded into the IR.

PC	**0**	3	0	0
MAR	**0**	3	0	0
MBR	**1**	3	0	4
IR	**1**	3	0	4
AC				



Opcodes

Load from Memory	1
Store to Memory	2
Add from Memory	3
Subtract from Memory	4

Memory

300	**1**	3	0	4
301	**3**	3	0	5
302	**2**	3	0	6
303	**.**	.	.	.
304	**0**	0	0	5
305	**0**	0	0	1
306				

Step 5 : Cycle 2

CPU Registers

That Instruction is over and the PC increments by 1

PC	**0**	3	0	*1*
MAR	**0**	3	0	0
MBR	**1**	3	0	4
IR	**1**	3	0	4
AC				

Address Bus

Opcodes

Load from Memory	1
Store to Memory	2
Add from Memory	3
Subtract from Memory	4

Memory

300	**1**	3	0	4
301	**3**	3	0	5
302	**2**	3	0	6
303	**.**	.	.	.
304	**0**	0	0	5
305	**0**	0	0	1
306				

Step 6

CPU Registers

The IR executes the instruction. The instruction tells the CPU to load memory #304 into the AC.

PC	**0**	3	0	1
MAR	**0**	3	0	0
MBR	**1**	3	0	4
IR	**1**	3	0	4
AC	**0**	0	0	5

Data Bus

Opcodes

Load from Memory	1
Store to Memory	2
Add from Memory	3
Subtract from Memory	4

Memory

300	**1**	3	0	4
301	**3**	3	0	5
302	**2**	3	0	6
303	**.**	.	.	.
304	**0**	0	0	5
305	**0**	0	0	1
306				

Step 7

CPU Registers

The PC changes the MAR to #301.

PC	**0**	3	0	1
MAR	**0**	3	0	1
MBR	**1**	3	0	4
IR	**1**	3	0	4
AC	**0**	0	0	5

Opcodes

Load from Memory	1
Store to Memory	2
Add from Memory	3
Subtract from Memory	4

Memory

300	**1**	3	0	4
301	**3**	3	0	5
302	**2**	3	0	6
303	**.**	.	.	.
304	**0**	0	0	5
305	**0**	0	0	1
306				

Step 8

CPU Registers

The MAR changing means that the MBR gets updated to 3305.

PC	**0**	3	0	1
MAR	**0**	3	0	1
MBR	**3**	3	0	5
IR	**1**	3	0	4
AC	**0**	0	0	5

Opcodes

Load from Memory	1
Store to Memory	2
Add from Memory	3
Subtract from Memory	4

Memory

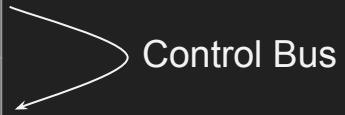
300	**1**	3	0	4
301	**3**	3	0	5
302	**2**	3	0	6
303	**.**	.	.	.
304	**0**	0	0	5
305	**0**	0	0	1
306				

Step 9

CPU Registers

The MBR contains an instruction so the IR gets changed to the value of the MBR

PC	**0**	3	0	1
MAR	**0**	3	0	1
MBR	**3**	3	0	5
IR	**3**	3	0	5
AC	**0**	0	0	5



Opcodes

Load from Memory	1
Store to Memory	2
Add from Memory	3
Subtract from Memory	4

Memory

300	**1**	3	0	4
301	**3**	3	0	5
302	**2**	3	0	6
303	**.**	.	.	.
304	**0**	0	0	5
305	**0**	0	0	1
306				

Step 10 : Cycle 3

CPU Registers

The IR is now executed.
It tells the CPU to **ADD** memory #305 to the AC.
This results in a value of 6 in the AC.

PC	**0**	3	0	1
MAR	**0**	3	0	1
MBR	**3**	3	0	5
IR	**3**	3	0	5
AC	**0**	0	0	6

Data Bus

Opcodes

Load from Memory	1
Store to Memory	2
Add from Memory	3
Subtract from Memory	4

Memory

300	**1**	3	0	4
301	**3**	3	0	5
302	**2**	3	0	6
303	**.**	.	.	.
304	**0**	0	0	5
305	**0**	0	0	1
306				

Step 11

CPU Registers

Now that we have the desired value in the AC, we need to push it to a Memory location. Using our 2 opcode that's in memory #306.

PC	**0**	3	0	1
MAR	**0**	3	0	1
MBR	**3**	3	0	5
IR	**3**	3	0	5
AC	**0**	0	0	6

Opcodes

Load from Memory	1
Store to Memory	2
Add from Memory	3
Subtract from Memory	4

Memory

300	**1**	3	0	4
301	**3**	3	0	5
302	**2**	3	0	6
303	**.**	.	.	.
304	**0**	0	0	5
305	**0**	0	0	1
306				

Step 12

CPU Registers

The PC increments again to #302, this changes the MAR in turn.

PC	**0**	3	0	2
MAR	**0**	3	0	2
MBR	**3**	3	0	5
IR	**3**	3	0	5
AC	**0**	0	0	6

Address Bus

Control Bus

Opcodes

Load from Memory	1
Store to Memory	2
Add from Memory	3
Subtract from Memory	4

Memory

300	**1**	3	0	4
301	**3**	3	0	5
302	**2**	3	0	6
303	**.**	.	.	.
304	**0**	0	0	5
305	**0**	0	0	1
306				

Step 13

CPU Registers

The MAR points to #302, and so memory #302 is loaded into the MBR. The new MBR value changes the IR.

PC	**0**	3	0	2
MAR	**0**	3	0	2
MBR	**2**	3	0	6
IR	**2**	3	0	6
AC	**0**	0	0	6

Data Bus

Opcodes

Load from Memory	1
Store to Memory	2
Add from Memory	3
Subtract from Memory	4

Memory

300	**1**	3	0	4
301	**3**	3	0	5
302	**2**	3	0	6
303	**.**	.	.	.
304	**0**	0	0	5
305	**0**	0	0	1
306				

Step 14

CPU Registers

The instruction in the IR gets executed. It's opcode 2 with a target of #306 so it stores the AC in memory #206.

PC	**0**	3	0	2
MAR	**0**	3	0	2
MBR	**2**	3	0	6
IR	**2**	3	0	6
AC	**0**	0	0	6

Data Bus

Opcodes

Load from Memory	1
Store to Memory	2
Add from Memory	3
Subtract from Memory	4

Memory

300	**1**	3	0	4
301	**3**	3	0	5
302	**2**	3	0	6
303	**.**	.	.	.
304	**0**	0	0	5
305	**0**	0	0	1
306	**0**	0	0	6

Step 15 : Cycle 4

CPU Registers

We have now finished and we have the desired value in #306.

PC	**0**	3	0	2
MAR	**0**	3	0	2
MBR	**2**	3	0	6
IR	**2**	3	0	6
AC	**0**	0	0	6

Opcodes

Load from Memory	1
Store to Memory	2
Add from Memory	3
Subtract from Memory	4

Memory

300	**1**	3	0	4
301	**3**	3	0	5
302	**2**	3	0	6
303	**.**	.	.	.
304	**0**	0	0	5
305	**0**	0	0	1
306	**0**	0	0	6