The OK-Computer

Julian Lore
Sandrine Monfourny-Daigneault
Jacques Vincent

April 10th, 2017

1 Assembler Program

What this program does:
Multiplies (using adder in a loop) a number from RAM and a number from input buffer (keypad), answer displayed on display
Assume 1111 is the RAM value that stores the number we want to multiply by

RAM Adress	RAM contents	Description
0000	0111 1000	# Load input into D1
0001	0101 0111	# If D0 isn't 0, go to prog loop
0010	1000 0000	# STOP, ends execution of program
0011	0000 0000	# Will store counter the we'll increment
0100	0000 0000	# Where we'll store answer
0101	0000 0110	# Number we want to multiply by, 6 as example
0110	0000 0001	# 1 for incrementing

Prog:

RAM Adress	RAM contents	Description
0111	1010 0100	# Load current answer into D0
1000	1011 0101	# Load number we want to multiply by into D1
1001	0010 0000	# D0=D0+D1
1010	0110 0000	# Prints current answer
1011	1110 0100	# Store D0 into current answer
1100	1010 0011	# Load counter into D0
1101	1011 0110	# Load one into D1
1110	0010 0000	# D0=D0+D1, increment counter
1111	1110 0011	# Store D0 into counter

Rely on natural looping of CPU

2 Description of Instructions

1. (CYCLE)

000
Op-Code
0-2

Not a real instruction (in the sense that it isn't a part of any program), it's what the CU uses to prepare the next instruction in RAM

2. LOAD: LD

101	x	xxxx
Op-Code	Register	Address
0-2	3	4-7

The address is sent to the MAR, which is then sent to the AR. The MODE is set to read mode and the contents located at the specified address are sent to the DR, then to the MBR. Then, the contents of the MBR are sent to the specified register.

3. STORE: STR

111	x	xxxx
Op-Code	Register	Address
0-2	3	4-7

The contents of the specified register are sent to the MBR. The address is sent to the MAR, which is then sent to the AR. The MODE is set to write mode and the contents of the MBR are sent to the DR, which then sends the contents to the byte of RAM at the specified address.

4. ADD: ADD

0010	x
Op-Code	Register to store into
0-3	4

The contents of D0 are sent to the ALU through the bus. The contents of D1 are also sent to the ALU through the bus. For both of these, the CU activates the output mode on D0 and D1. Then, the ALU is set to add mode and the addition is done. Then, the result is sent to D0 through the bus and by activating the input mode on D0.

5. SUBTRACT: SUB

0011	x
Op-Code	Register to store into
0-3	4

The contents of D0 are sent to the ALU through the bus. The contents of D1 are also sent to the ALU through the bus. For both of these, the CU activates the output mode on D0 and D1. Then, the ALU is set to subtract mode and the addition is done. Then, the result is sent to D0 through the bus and by activating the input mode on D0.

6. Branch equal: BEQ

0100	x
Op-Code	Address
0-3	4-7

The contents of D0 and D1 are directly sent to a comparison checker. When they are equal, it outputs TRUE, and this output is sent to the PC. When/if it receives this TRUE signal, it will load the adress specified.

7. Branc not equal: BNQ

0101	x
Op-Code	Address
0-3	4-7

The contents of D0 and D1 are directly sent to a comparison checker. When they are not equal, it outputs TRUE, and this output is sent to the PC. When/if it receives this TRUE signal, it will load the adress specified.

8. PRINT: PRT

0110	x
Op-Code	Register to display
0-3	4

The contents of the specified register are sent to the MBR through the bus, which are then sent to the Display Companion Register through the bus. Then, the display will show whatever is stored in the register.

9. INPUT: INP

0111	x
Op-Code	Register to store into
0-3	4

The input is converted to binary and sent to the MBR through the bus, which is then sent to the specified register through the bus.

10. STOP: STOP

1000
Op-Code
0-3

Permanently activates a pin in the CPU, which prevents the CYCLE command from being executed and therefore stops the program.

11. MULTIPLICATION: MULT

1001	x	xxx
Op-Code	Register to store into	Constant
0-3	4	5-7