8-Bit CPU Manual

Breadboard Computing

Contents

1 Overview

2 Processor Architecture

2.1 Registers:

- ◇ A Register: `REG_A_0`, `REG_A_1`, . . . , `REG_A_7`
- ◇ B Register: `REG_B_0`, `REG_B_1`, . . . , `REG_B_7`
- ◇ Temporary Register: `REG_TMP_00`, `REG_TMP_01`, . . . , `REG_TMP_15`
- ◇ Memory Address Register: `REG_MAR_00`, `REG_MAR_01`, . . . , `REG_MAR_15`
- ◇ Memory Buffer Register: `REG_MBR_0`, `REG_MBR_1`, . . . , `REG_MBR_7`
- ◇ Flag Register: `REG_F_0`, `REG_F_1`, . . . , `REG_F_7`
- ◇ Instruction Register: `REG_IR_0`, `REG_IR_1`, . . . , `REG_IR_7`

2.2 Buses

- ◇ Data Bus: `BUS_0`, `BUS_1`, . . . , `BUS_7`
- ◇ Address Bus: `ADDRESS_00`, `ADDRESS_01`, . . . , `ADDRESS_15`

2.3 Flags

- ◇ REG_F_0: sign flag(1 → result is negative; 0 → result is non-negative)
- ◇ REG_F_1: parity flag(1 → result is odd; 0 → result is even)
- ◇ REG_F_2: zero flag(1 → result is not equal to zero; 0 → result is equal to zero)
- ◇ REG_F_3: carry flag(for unsigned number operations: 1 → result is out of range; 0 → result is correct)
- ◇ REG_F_4: overflow flag(for signed number operations: 1 → result is out of range; 0 → result is correct)

2.4 Signals

Clock			
CLK		signal generated by clock with specified frequency	
$\overline{\text{CLK}}$		inverted signal generated by clock with specified frequency	

Table 1: Clock Signals Description

Arithmetic Logic Unit			
REG_A_LOAD	A0	load data from data bus to register A	combined with CLK
REG_B_LOAD	A1	load data from data bus to register B	combined with CLK
ALU_OPC_0	A2	bit 0 of operation code to perform	
ALU_OPC_1	A3	bit 1 of operation code to perform	
ALU_OPC_2	A4	bit 2 of operation code to perform	
ALU_OPC_3	A5	bit 3 of operation code to perform	
ALU_OPC_4	A6	bit 4 of operation code to perform	
REG_F_LOAD	A7	load flags of current operation to flag register	combined with CLK
$\overline{\text{REG\_F\_OUT}}$	B0	pass data from flag register to data bus	
$\overline{\text{ALU\_OUT}}$	B1	pass current operation result to data bus	

Table 2: Arithmetic Logic Unit Signals Description

2.5 Instructions

2.6 Instruction Cycle

- ◇ Fetch(fixed number of cycles)

Data Bus - Address Bus Connector			
REG_TMPH_LOAD	B2	load data from data/address bus to higher bits of temporary register	combined with CLK
REG_TMPL_LOAD	B3	load data from data/address bus to lower bits of temporary register	combined with CLK
$\overline{\text{REG\_TMPH\_OUT}}$	B4	pass data from higher bits of temporary register to data/address bus	
$\overline{\text{REG\_TMPL\_OUT}}$	B5	pass data from lower bits of temporary register to data/address bus	
$\overline{\text{REG\_TMP\_PASS\_ADDRESS}}$	B6	enable passing address from address bus/temporary register to temporary register/address bus	
$\overline{\text{REG\_TMPH\_PASS\_DATA}}$	B7	enable passing data from data bus/msb of temporary register to msb of temporary register/data bus	
$\overline{\text{REG\_TMPL\_PASS\_DATA}}$	C0	enable passing data from data bus/lsb of temporary register to lsb of temporary register/data bus	
REG_TMP_ADDRESS_DIR	C1	address direction selector: from address bus/temporary register to temporary register/address bus	
REG_TMPH_DATA_DIR	C2	data flow direction selector: from data bus/msb of temporary register to msb of temporary register/data bus	
REG_TMPL_DATA_DIR	C3	data flow direction selector: from data bus/lsb of temporary register to lsb temporary register/data bus	

Table 3: Data Bus - Address Bus Connector Signals Description

Program Counter			
$\overline{\text{PC\_LOAD}}$	C4	load address from address bus to program counter	
$\overline{\text{PC\_RST}}$	C5	set program counter to 0x0000	
PC_TICK	C6	increment program counter	combined with $\overline{\text{CLK}}$
$\overline{\text{PC\_OUT}}$	C7	pass data from program counter to address bus	

Table 4: Program Counter Signals Description

- ◇ Decode(variable number of cycles)
- ◇ Execute(variable number of cycles)

2.7 Decoder

Input: $MI_7I_6I_5I_4I_3I_2I_1I_0C_3C_2C_1C_0$:

- ◇ M - operating mode,
- ◇ $I_7I_6I_5I_4I_3I_2I_1I_0$ - instruction,

Stack Counter			
$\overline{\texttt{STC\_LOAD}}$	D0	load address from address bus to stack counter	
STC_RST	D1	set stack counter to 0x0000	
STC_TICK	D2	increment/decrement stack counter; increment/decrement depends on mode	combined with $\overline{\texttt{CLK}}$
STC_MODE	D3	stack counter operation selector: increment or decrement	
$\overline{\texttt{STC\_OUT}}$	D4	pass data from stack counter to address bus	

Table 5: Stack Counter Signals Description

Memory Unit			
REG_MAR_LOAD	D5	load data from address bus to memory address register	combined with CLK
REG_MBR_LOAD	D6	load data from data bus to memory buffer register	combined with CLK
$\overline{\texttt{MEM\_OUT}}$	D7	read data to memory buffer register from memory at address from memory address register	
$\overline{\texttt{MEM\_IN}}$	E0	write data from memory buffer register to memory at address from memory address register	
MEM_PART	E1	additional bit of memory address; it is used to easy memory division	
$\overline{\texttt{ZERO\_PAGE}}$	E2	clear higher bits of memory address; it is used to faster access to memory at low addresses	
REG_MBR_WORD_DIR	E3	data flow direction selector: pass data from data bus/memory buffer register to memory buffer register/data bus	
$\overline{\texttt{REG\_MAR\_USE\_BTTNS}}$	E4	use buttons to enter value to set to memory address register	
$\overline{\texttt{REG\_MBR\_USE\_BTTNS}}$	E5	use buttons to enter value to set to memory buffer register	
$\overline{\texttt{REG\_MBR\_USE\_BUS}}$	E6	use bus to pass value to set to memory buffer register or get value from memory buffer register	

Table 6: Memory Unit Signals Description

- $C_3C_2C_1C_0$ - microcode counter

Control Unit			
IR_LOAD	F0	load instruction from data bus	combined with $\overline{\texttt{CLK}}$
MCC_TICK	F1	increment microcode counter	combined with $\overline{\texttt{CLK}}$
$\overline{\texttt{MCC\_RST}}$	F2	set microcode counter to 0	

Table 7: Control Unit Signals Description