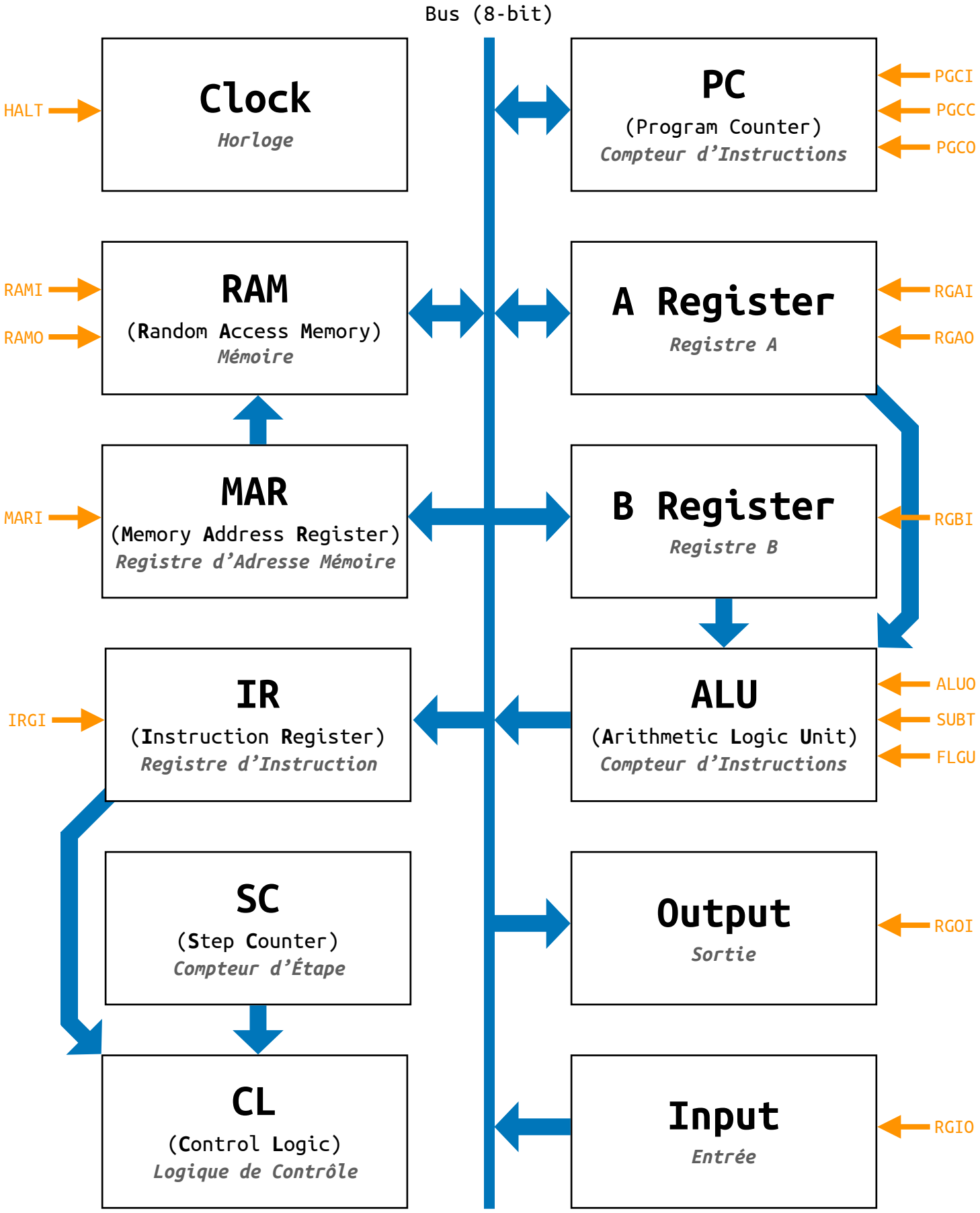
Architecture

GilDev Breadboard Computer



Instruction Set

GilDev Breadboard Computer

Instruction	Opcode	Size (bytes)	Update flags	Meaning	Description
NOP	**0000**	1	No	**No OP**eration	Do nothing
LDA	**0001**	2	No	**LoaD A**ddress	Load memory content at address given by operand byte in accumulator
LDI	**0010**	2	No	**LoaD I**mmediate	Load operand byte in accumulator
STA	**0011**	2	No	**ST**ore **A**ddress	Store content of accumulator in memory at address given by operand byte
ADD	**0100**	2	Yes	**ADD**	Add memory content at address given by operand byte to accumulator
ADI	**0101**	2	Yes	**AD**d **I**mmediate	Add operand byte to accumulator
SUB	**0110**	2	Yes	**SUB**stract	Subtract memory content at address given by operand byte from accumulator
SUI	**0111**	2	Yes	**SU**bstract **I**mmediate	Subtract operand byte from accumulator
JMP	**1000**	2	No	**JuMP**	Jump to address given by operand byte
JPN	**1001**	2	No	**JumP N**egative	Jump to address given by operand byte if N flag is set
JPP	**1010**	2	No	**JumP P**ositive	Jump to address given by operand byte if N flag and Z flag are not set
JPC	**1011**	2	No	**JumP C**arry	Jump to address given by operand byte if C flag is set
JPZ	**1100**	2	No	**JumP Z**ero	Jump to address given by operand byte if Z flag is set
OUT	**1101**	1	No	**OUT**put	Show content of accumulator on the display
INP	**1110**	1	No	**INP**ut	Load input number in accumulator
HLT	**1111**	1	No	**HaLT**	Stops the clock

Control Signals

GilDev Breadboard Computer

Name	Meaning	Description
HALT	**HALT**	Stops the clock
FLGU	**FL**a**G**s **U**pdate	Update flags register using ALU's current output
MARI	Memory **A**ddress **R**egister **I**n	Load from bus into MAR
IRGI	**I**nstruction **R**e**G**ister **I**n	Load from bus into IR
RGAI	**R**e**G**ister **A** **I**n	Load from bus into register A (accumulator)
RGAO	**R**e**G**ister **A** **O**ut	Put content of register A (accumulator) onto the bus
RGBI	**R**e**G**ister **B** **I**n	Load from bus into register B
RGOI	**R**e**G**ister **O**utput **I**n	Load from bus into register Output
RGIO	**R**e**G**ister **I**nput **O**ut	Put content of input onto the bus
RAMI	**R**andom **A**ccess **M**emory **I**n	Load from bus into memory at address pointed by the MAR
RAMO	**R**andom **A**ccess **M**emory **O**ut	Put memory content at address pointed by the MAR onto the bus
ALUO	**A**rithmetic **L**ogic **U**nit **O**ut	Put sum result of ALU onto the bus
SUBT	**SUBT**ract	Invert and increment second input of ALU in order to subtract B from A
PGCI	**P**ro**G**ram **C**ounter **I**n	Load from bus into PC
PGCO	**P**ro**G**ram **C**ounter **O**ut	Put content of PC onto the bus
PGCC	**P**ro**G**ram **C**ounter **C**ount	Increment PC

Microcode

GilDev Breadboard Computer

Instruction	Opcode	Step	Flags (NCZ)	HALT	FLGU	MARI	IRGI	RGAI	RGAO	RGBI	RGOI	RGIO	RAMI	RAMO	ALUO	SUBT	PGCI	PGCO	PGCC
Fetch	XXXX	000	xxx	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0
		001		0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	1
NOP	0000	010	xxx	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
LDA	0001	010	xxx	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0
		011		0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	1
		100		0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0
LDI	0010	010	xxx	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0
		011		0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	1
STA	0011	010	xxx	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0
		011		0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	1
		100		0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0
ADD	0100	010	xxx	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0
		011		0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	1
		100		0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0
		101		0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		110		0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0
ADI	0101	010	xxx	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0
		011		0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	1
		100		0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		101		0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0
SUB	0110	010	xxx	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0
		011		0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	1
		100		0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0
		101		0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0
		110		0	0	0	0	1	0	0	0	0	0	0	1	1	0	0	0
SUI	0111	010	xxx	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0
		011		0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	1
		100		0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0
		101		0	0	0	0	1	0	0	0	0	0	0	1	1	0	0	0
JMP	1000	010	xxx	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0
				0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0
JPN	1001	010	0xx	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
		010	1xx	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0
		011	1xx	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0
JPP	1010	010	xxx	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
		010	0x0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0
		011	0x0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0
JPC	1011	010	x0x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
		010	x1x	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0
		011	x1x	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0
JPZ	1100	010	xx0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
		010	xx1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0
		011	xx1	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0
OUT	1101	010	xxx	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0
INP	1110	010	xxx	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0
HLT	1111	010	xxx	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Instruction	Opcode	Step	Flags (NCZ)	HALT	FLGU	MARI	IRGI	RGAI	RGAO	RGBI	RGOI	RGIO	RAMI	RAMO	ALUO	SUBT	PGCI	PGCO	PGCC

Programs

GilDev Breadboard Computer

Adder

ADDRESS	CONTENT	LABEL	INSTRUCTION	COMMENT
0000 0000	1110	start	INP	a = input()
0000 0001	0011		STA a	
0000 0010	1000 0000			
0000 0011	1110		INP	b = input()
0000 0100	0100		ADD a	a += b
0000 0101	1000 0000			
0000 0110	1101		OUT	output(a)
0000 0111	1000		JMP start	goto start
0000 1000	0000			
1000 0000		a		

Subtractor

ADDRESS	CONTENT	LABEL	INSTRUCTION	COMMENT
0000 0000	1110	start	INP	a = input()
0000 0001	0011		STA a	
0000 0010	1000 0000			
0000 0011	1110		INP	b = input()
0000 0100	0110		SUB a	a -= a b
0000 0101	1000 0000			
0000 0110	1101		OUT	output(a)
0000 0111	1000		JMP start	goto start
0000 1000	0000			
1000 0000		a		

Countdown until 0

ADDRESS	CONTENT	LABEL	INSTRUCTION	COMMENT
0000 0000	1110	start	INP	a = input()
0000 0001	0111	loop	SUI #1	input--
0000 0010	0001			
0000 0011	1101		OUT	output(a)
0000 0100	1100		JPZ start	if (a == 0)
0000 0101	0000			goto start
0000 0110	1000		JMP loop	goto loop
0000 0111	0001			

Multiplication tables from 1 to 10

ADDRESS	CONTENT	LABEL	INSTRUCTION	COMMENT
0000 0000	0010	start	LDI #1	
0000 0001	0001			
0000 0010	0011		STA inc	inc = 1
0000 0011	1000 0000			
0000 0100	1101	loop	OUT	output(inc)
0000 0101	0100		ADD inc	inc *= 2
0000 0110	1000 0000			
0000 0111	1011		JPC next	if (inc > 255)
0000 1000	1011			goto next
0000 1001	1000		JMP loop	goto loop
0000 1010	0100			
0000 1011	0001	next	LDA inc	
0000 1100	1000 0000			
0000 1101	0101		ADI #1	inc++
0000 1110	0001			
0000 1111	0011		STA inc	
0001 0000	1000 0000			
0001 0001	0111		SUI #11	*Table up to 10*
0001 0010	1011			
0001 0011	1100		JPZ start	if (inc == 11)
0001 0100	0000			goto start
0001 0101	0001		LDA inc	
0001 0110	1000 0000			
0001 0111	1000		JMP loop	goto loop
0001 1000	0100			
1000 000		inc		

Multiplier

ADDRESS	CONTENT	LABEL	INSTRUCTION	COMMENT
0000 0000	1110	start	INP	a = input()
0000 0001	0011		STA total	total = a
0000 0010	1000 0001			
0000 0011	0011		STA amount	amount = a
0000 0100	1000 0010			
0000 0101	1110		INP	
0000 0110	0011		STA i	i = input()
0000 0111	1000 0000			
0000 1000	0001	loop	LDA i	
0000 1001	1000 0000			
0000 1010	0111		SUI #1	i--
0000 1011	0001			
0000 1100	1100		JEZ end	if (i == 0)
0000 1101	0001 1000			goto end
0000 1110	0011		STA i	
0000 1111	1000 0000			
0001 0000	0001		LDA total	
0001 0001	1000 0001			
0001 0010	0100		ADD amount	total += amount
0001 0011	1000 0010			
0001 0100	0011		STA total	
0001 0101	1000 0001			
0001 0110	1000		JMP loop	goto loop
0001 0111	0000 1000			
0001 1000	0001	end	LDA total	
0001 1001	1000 0001			
0001 1010	1101		OUT	output(total)
0001 1011	1111		HLT	
1000 0000		i		
1000 0001		total		
1000 0010		amount		