

CPU UNUS-8S

Definições de Projeto

Largura da arquitetura: 8 *bits* (1 *byte*)

Número de células endereçáveis: 256

Número de registradores de propósito geral: 0 (todas as operações acontecem na pilha)

Registrador de propósito específico para pilha (SP), aponta para o último elemento da pilha

Número de endereços (operandos) por operação: 0 endereços

Modos de endereçamento explícitos: 2 (imediato e direto)

Instruções de tamanho variável: 1 ou 2 bytes

Linguagem Assembly e Código ISA

Instrução	Código ISA	Descrição
NOP	0x00	Não faz nada
PUSHI imediato	0x10 0xNN	MEM[ToS] \leftarrow imediato; INC ToS
PUSH endereço	0x11 0xNN	MEM[ToS] \leftarrow MEM[endereço]; INC ToS
POP endereço	0x20 0xNN	MEM[endereço] \leftarrow MEM[ToS]; DEC ToS
ADD	0x40	MEM[ToS-1] \leftarrow MEM[ToS] + MEM[Tos-1]; DEC Tos
AND	0x41	MEM[ToS-1] \leftarrow MEM[ToS] AND MEM[Tos-1]; DEC Tos
OR	0x42	MEM[ToS-1] \leftarrow MEM[ToS] OR MEM[Tos-1]; DEC Tos
NOT	0x53	MEM[ToS] \leftarrow NOT MEM[ToS]
INC	0x54	MEM[ToS] \leftarrow INC MEM[ToS]
SRL	0x55	MEM[ToS] \leftarrow MEM[ToS] \gg 1
SLL	0x56	MEM[ToS] \leftarrow MEM[ToS] \ll 1
SRA	0x57	MEM[ToS] \leftarrow MEM[ToS] \gg 1 (sinal preservado)
DUP	0x60	MEM[ToS+1] \leftarrow MEM[ToS]; INC ToS
JN endereço	0xA0 0xNN	SE N PC \leftarrow imediato
JZ endereço	0xA1 0xNN	SE Z PC \leftarrow imediato
JC endereço	0xA2 0xNN	SE C PC \leftarrow imediato
JMP endereço	0xAF 0xNN	PC \leftarrow imediato

Circuitos

Os desenhos preliminares dos circuitos da CPU UNUS-8S estão descritos em CPUv2 e CTRLv4 no arquivo UNUS-8S.v1.circ.

Unidade de Controle

A unidade de controle é uma máquina de estados com uma entrada de 8 *bits* do sinal *Instr*, onde os estados foram abstraídos para dar lugar a um registrador (uPC) que contém o endereço da linha atual da ROM de micro-códigos. O valor de uPC pode ser incrementado (*Next*), colocado em zero (*Fetch*) ou receber um endereço da primeira micro-instrução relativa à instrução corrente (*Dispatch*), vindo de um tradutor que converte o código da operação (*opcode*) em micro-endereço.

A implementação da máquina de estados é feita através de uma memória ROM com os *bits* das micro-instruções e os *bits* do uJump. Essa memória apresenta 256 linhas e o conteúdo de 20 *bits* por célula contendo todos os sinais para o *datapath* (MEMrd, MEMwr, REGwr, Asel, Bsel, Csel, ALUop), os sinais consumidos internamente (FLAGSwr, Halt) e o uJump (3 *bits*), conforme a tabela a seguir.

uJump	Valor	uJump	Valor
Next	000	CondN	100
Fetch	001	CondZ	101
Dispatch	010	CondC	110
N.U.	011	CondV	111

Banco de Registradores

Os registradores presentes na arquitetura, em sua maioria invisíveis, são: ZERO (000), MDR (001), MAR (010), IR (011), PC (100), BP (101), AUX(110) e SP (111). Todos os registradores podem ter seus conteúdos passados para os barramentos internos ABus e BBus e podem ser escritos pelo conteúdo presente no CBus. O registrador ZERO não pode ser escrito e seu valor é sempre zero.

Microprograma

O micro-código preliminar para as instruções estão descritas a seguir:

	μInstrução	uJump	Pseudocódigo
fetch0	REGwr; Asel=PC; Bsel=ZERO; Csel=MAR; ALUop=OR	Next	MAR ← PC
fetch1	MEMrd; REGwr; Asel=PC; Csel=PC; ALUop=INC	Next	MDR ← MEM PC ← PC + 1
fetch2	REGwr; Asel=MDR; Bsel=ZERO; Csel=IR; ALUop=OR	Dispatch	IR ← MDR
nop0		Fetch	
push0	REGwr; Asel=PC; Bsel=ZERO; Csel=MAR; ALUop=OR	Next	MAR ← PC
push1	MEMrd; REGwr; Asel=PC; Csel=PC; ALUop=INC	Next	MDR ← MEM PC ← PC + 1
push2	REGwr; Asel=AUX; Csel=AUX; ALUop=NOT	Next	AUX ← -1
push3	MEMrd; REGwr; Asel=PC; Csel=PC; ALUop=INC	Next	SP ← SP + AUX
push4	REGwr; Asel=SP; Bsel=ZERO; Csel=MAR; ALUop=OR	Next	MAR ← SP

	μInstrução	uJump	Pseudocódigo
push5	MEMw _r	Fetch	MEM ← MDR
pushi0	REGw _r ; Asel=PC; Bsel=ZERO; CSel=MAR; ALUop=OR	Next	MAR ← PC
pushi1	MEMr _d ; REGw _r ; Asel=PC; CSel=PC; ALUop=INC	Next	MDR ← MEM PC ← PC + 1
pushi2	REGw _r ; Asel=AUX; CSel=AUX; ALUop=NOT	Next	AUX ← -1
pushi3	REGw _r ; Asel=SP; Bsel=AUX; CSel=SP; ALUop=ADD	Next	SP ← SP + AUX
pushi4	REGw _r ; Asel=SP; Bsel=ZERO; CSel=MAR; ALUop=OR	Next	MAR ← SP
pushi5	MEMw _r ;	Fetch	MEM ← MDR
pop0	REGw _r ; Asel=SP; Bsel=ZERO; CSel=MAR; ALUop=OR	Next	MAR ← SP
pop1	MEMr _d ; REGw _r ; Asel=SP; CSel=SP; ALUop=INC	Next	MDR ← MEM SP ← SP + 1
pop2	REGw _r ; Asel=PC; Bsel=ZERO; CSel=MAR; ALUop=OR	Next	MAR ← PC
pop3	MEMr _d ; REGw _r ; Asel=PC; Bsel=ZERO; CSel=PC; ALUop=OR	Next	AUX ← MDR MDR ← MEM
pop4	REGw _r ; Asel=MDR; Bsel=ZERO; CSel=AUX; ALUop=OR	Next	MAR ← MDR
pop5	REGw _r ; Asel=AUX; Bsel=ZERO; CSel=MDR; ALUop=OR	Next	MDR ← AUX
pop6	MEMw _r	Fetch	MEM ← MDR
add0	REGw _r ; Asel=SP; Bsel=ZERO; CSel=MAR; ALUop=OR	Next	MAR ← SP
add1	MEMr _d ; REGw _r ; Asel=SP; CSel=SP; ALUop=INC	Next	MDR ← MEM SP ← SP + 1
add2	REGw _r ; Asel=SP; Bsel=ZERO; CSel=MAR; ALUop=OR	Next	MAR ← SP
add3	MEMr _d ; REGw _r ; Asel=MDR; Bsel=ZERO; CSel=AUX; ALUop=OR	Next	AUX ← MDR MDR ← MEM
add4	REGw _r ; Asel=MDR; Bsel=AUX; CSel=MDR; ALUop=ADD; FLAGSw_r	Next	MDR ← MDR + AUX
add5	MEMw _r	Fetch	MEM ← MDR
and0	REGw _r ; Asel=SP; Bsel=ZERO; CSel=MAR; ALUop=OR	Next	MAR ← SP
and1	MEMr _d ; REGw _r ; Asel=SP; CSel=SP; ALUop=INC	Next	MDR ← MEM SP ← SP + 1
and2	REGw _r ; Asel=SP; Bsel=ZERO; CSel=MAR; ALUop=OR	Next	MAR ← SP
and3	MEMr _d ; REGw _r ; Asel=MDR; Bsel=ZERO; CSel=AUX; ALUop=OR	Next	AUX ← MDR MDR ← MEM
and4	REGw _r ; Asel=MDR; Bsel=AUX; CSel=MDR; ALUop=AND; FLAGSw_r	Next	MDR ← MDR + AUX
and5	MEMw _r	Fetch	MEM ← MDR
or0	REGw _r ; Asel=SP; Bsel=ZERO; CSel=MAR; ALUop=OR	Next	MAR ← SP
or1	MEMr _d ; REGw _r ; Asel=SP; CSel=SP; ALUop=INC	Next	MDR ← MEM SP ← SP + 1

	μInstrução	uJump	Pseudocódigo
or2	REGwr; Asel=SP; Bsel=ZERO; CSel=MAR; ALUop=OR	Next	MAR ← SP
or3	MEMrd; REGwr; Asel=MDR; Bsel=ZERO; CSel=AUX; ALUop=OR	Next	AUX ← MDR MDR ← MEM
or4	REGwr; Asel=MDR; Bsel=AUX; CSel=MDR; ALUop=OR; FLAGSwr	Next	MDR ← MDR + AUX
or5	MEMwr	Fetch	MEM ← MDR
not0	REGwr; Asel=SP; Bsel=ZERO; CSel=MAR; ALUop=OR	Next	MAR ← SP
not1	MEMrd;	Next	MDR ← MEM
not2	REGwr; Asel=MDR; CSel=MDR; ALUop=NOT; FLAGSwr	Next	MDR ← NOT MDR
not3	MEMwr	Fetch	MEM ← MDR
inc0	REGwr; Asel=SP; Bsel=ZERO; CSel=MAR; ALUop=OR	Next	MAR ← SP
inc1	MEMrd;	Next	MDR ← MEM
inc2	REGwr; Asel=MDR; CSel=MDR; ALUop=INC; FLAGSwr	Next	MDR ← INC MDR
inc3	MEMwr	Fetch	MEM ← MDR
srl0	REGwr; Asel=SP; Bsel=ZERO; CSel=MAR; ALUop=OR	Next	MAR ← SP
srl1	MEMrd;	Next	MDR ← MEM
srl2	REGwr; Asel=MDR; CSel=MDR; ALUop=SRL; FLAGSwr	Next	MDR ← SRL MDR
srl3	MEMwr	Fetch	MEM ← MDR
sll0	REGwr; Asel=SP; Bsel=ZERO; CSel=MAR; ALUop=OR	Next	MAR ← SP
sll1	MEMrd;	Next	MDR ← MEM
sll2	REGwr; Asel=MDR; CSel=MDR; ALUop=SLL; FLAGSwr	Next	MDR ← SLL MDR
sll3	MEMwr	Fetch	MEM ← MDR
sra0	REGwr; Asel=SP; Bsel=ZERO; CSel=MAR; ALUop=OR	Next	MAR ← SP
sra1	MEMrd;	Next	MDR ← MEM
sra2	REGwr; Asel=MDR; CSel=MDR; ALUop=SRA; FLAGSwr	Next	MDR ← SRA MDR
sra3	MEMwr	Fetch	MEM ← MDR
dup0	REGwr; Asel=SP; Bsel=ZERO; CSel=MAR; ALUop=OR	Next	MAR ← PC
dup1	MEMrd; REGwr; Asel=AUX; CSel=AUX; ALUop=NOT	Next	MDR ← MEM AUX ← -1
dup2	REGwr; Asel=SP; Bsel=AUX; CSel=SP; ALUop=ADD	Next	SP ← SP + AUX
dup3	REGwr; Asel=SP; Bsel=ZERO; CSel=MAR; ALUop=OR	Next	MAR ← SP
dup4	MEMwr	Fetch	MEM ← MDR
jn0	REGwr; Asel=PC; Bsel=ZERO; CSel=MAR; ALUop=OR	Next	MAR ← PC

	μInstrução	uJump	Pseudocódigo
jn1	MEMrd; REGwr; Asel=PC; CSel=PC; ALUop=INC	Next	MDR ← MEM PC ← PC + 1
jn2	REGwr; Asel=MDR; Bsel=ZERO; CSel=PC; ALUop=OR	CondN	PC ← MDR
jn3		Fetch	
jz0	REGwr; Asel=PC; Bsel=ZERO; CSel=MAR; ALUop=OR	Next	MAR ← PC
jz1	MEMrd; REGwr; Asel=PC; CSel=PC; ALUop=INC	Next	MDR ← MEM PC ← PC + 1
jz2	REGwr; Asel=MDR; Bsel=ZERO; CSel=PC; ALUop=OR	CondZ	PC ← MDR
jz3		Fetch	
jc0	REGwr; Asel=PC; Bsel=ZERO; CSel=MAR; ALUop=OR	Next	MAR ← PC
jc1	MEMrd; REGwr; Asel=PC; CSel=PC; ALUop=INC	Next	MDR ← MEM PC ← PC + 1
jc2	REGwr; Asel=MDR; Bsel=ZERO; CSel=PC; ALUop=OR	CondC	PC ← MDR
jc3		Fetch	
jc0	REGwr; Asel=PC; Bsel=ZERO; CSel=MAR; ALUop=OR	Next	MAR ← PC
jc1	MEMrd;	Next	MDR ← MEM
jc2	REGwr; Asel=MDR; Bsel=ZERO; CSel=PC; ALUop=OR	Fetch	PC ← MDR

uROM

	μInstrução							iSign		μJump	μROM	
	MEMrd	MEMwr	REGwr	Asel	Biel	Csel	ALUop	FLAGSwr	Halt		End.	Conteúdo
fetch0			1	100	000	010	010			0	00	002424
fetch1	1		1	100		100	100			0	01	004825
fetch2			1	001		011	010			2	02	04260C
nop0										1	03	020000
push0			1	100	000	010	010			0	04	002424
push1	1		1	100		100	100			0	05	004825
push2			1	110		110	011			0	06	003C34
push3	1		1	100		100	100			0	07	004825
push4			1	111	000	010	010			0	08	00243C
push5		1								1	09	020002
pushi0			1	100	000	010	010			0	0A	002424
pushi1	1		1	100		100	100			0	0B	004825
pushi2			1	110		110	011			0	0C	003C34
pushi3			1	111	110	111	000			0	0D	000FBC

	μInstrução							iSign		μJump	μROM	
	MEMrd	MEMwr	REGwr	Asel	Biel	Csel	ALUop	FLAGSwr	Halt		End.	Conteúdo
pushi4			1	111	000	010	010			0	0E	00243C
pushi5		1								1	0F	020002
pop0			1	111	000	010	010			0	10	00243C
pop1	1		1	111		111	100			0	11	004E3D
pop2			1	100	000	010	010			0	12	002424
pop3	1		1	100	000	100	010			0	13	002825
pop4			1	001	000	110	010			0	14	002C0C
pop5			1	110	000	001	010			0	15	002234
pop6		1								1	16	020002
add0			1	111	000	010	010			0	17	00243C
add1	1		1	111		111	100			0	18	004E3D
add2			1	111	000	010	010			0	19	00243C
add3	1		1	001	000	110	010			0	1A	002C0D
add4			1	001	110	001	000	1		0	1B	00838C
add5		1								1	1C	020002
and0			1	111	000	010	010			0	1D	00243C
and1	1		1	111		111	100			0	1E	004E3D
and2			1	111	000	010	010			0	1F	00243C
and3	1		1	001		110	010			0	20	002C0D
and4			1	001	110	001	001	1		0	21	00938C
and5		1								1	22	020002
or0			1	111	000	010	010			0	23	00243C
or1	1		1	111		111	100			0	24	004E3D
or2			1	111	000	010	010			0	25	00243C
or3	1		1	001	000	110	010			0	26	002C0D
or4			1	001	110	001	010	1		0	27	00A38C
or5		1								1	28	020002
not0			1	111	000	010	010			0	29	00243C
not1	1									0	2A	000001
not2			1	001		001	011	1		0	2B	00B20C
not3		1								1	2C	020002
inc0			1	111	000	010	010			0	2D	00243C
inc1	1									0	2E	000001

	µInstrução							iSign		µJump	µROM	
	MEMrd	MEMwr	REGwr	Asel	Biel	Csel	ALUop	FLAGSwr	Halt		End.	Conteúdo
inc2			1	001		001	100	1		0	2F	00C20C
inc3		1								1	30	020002
srl0			1	111	000	010	010			0	31	00243C
srl1	1									0	32	000001
srl2			1	001		001	101	1		0	33	00D20C
srl3		1								1	34	020002
slr0			1	111	000	010	010			0	35	00243C
slr1	1									0	36	000001
slr2			1	001		001	110	1		0	37	00E20C
slr3		1								1	38	020002
sra0			1	111	000	010	010			0	39	00243C
sra1	1									0	3A	000001
sra2			1	001		001	111	1		0	3B	00F20C
sra3		1								1	3C	020002
dup0			1	111	000	010	010			0	3D	00243C
dup1	1		1	110		110	011			0	3E	003C35
dup2			1	111	000	111	000			0	3F	000E3C
dup3			1	111	000	010	010			0	40	00243C
dup4		1								1	41	020002
jn0			1	100	000	010	010			0	42	002424
jn1	1		1	100		100	100			0	43	004825
jn2			1	001		100	010			4	44	08280C
jn3										1	45	020000
jz0			1	100	000	010	010			0	46	002424
jz1	1		1	100		100	100			0	47	004825
jz2			1	001	000	100	010			5	48	0A280C
jz3										1	49	020000
jc0			1	100	000	010	010			0	4A	002424
jc1	1		1	100		100	100			0	4B	004825
jc2			1	001	000	100	010			6	4C	0C280C
jc3										1	4D	020000
jmp0			1	100	000	010	010			0	4E	002424
jmp1	1									0	4F	000001

	μInstrução							iSign		μJump	μROM	
	MEMrd	MEMwr	REGwr	Asel	Biel	Csel	ALUop	FLAGSwr	Halt		End.	Conteúdo
jmp2			1	001	000	100	010			1	50	02280C