P8 Instruction Set

Transfer Instructions

JMP Jump to 10 bit address. Code must be SLOT1.

RET Subroutine return. Pop **R** stack to **P**.

JZ Jump if **T** is zero.

JNC Jump if CARRY is cleared.

CALL Call subroutine. Push **P** to **R** stack. Load **P** (bits 10..0) from **I**.

Memory Access Instructions

LDP Load memory[**A**] to **T**. Push **T** to **N** stack. Increment **A**.

LIT Load in-line literal to **T**. Push **T** to **N** stack.

LD Load memory[**A**] to **T**. Push **T** to **N** stack.

STP Store **T** to memory[**A**]. Pop **N** stack to **T**. Increment **A**.

ST Store **T** to memory[**A**]. Pop **N** stack to **T**.

ALU Instructions

COM Complement **T**.

SHL Shift **T** left 1 bit.

SHR Shift **T** right 1 bit.

ADDC if $\mathbf{T}_0$=1 then **N** + **T** => **T**. **N** preserved.

XOR Pop **N** stack. **N** XOR **T** => **T**.

AND Pop **N** stack. **N** AND **T** => **T**.

ADD Pop **N** stack. **N** + **T** => **T**.

Register Instructions

POP Push **T** to **N** stack. Pop **R** stack to **T.**

LDA Push **T** to **N** stack. Load **A** to **T**.

DUP Push **T** to **N** stack. Duplicates **T**.

PUSH Push **T** to **R** stack. Pop **N** stack to **T**.

STA Store **T** to **A**. Pop **N** stack to **T**.

NOP No operation.

DROP Pop **N** stack to **T**.

Register names are **CAPITALIZED BOLD**.

Instruction Fetch occurs during PHASE0. **I** register is enabled when PHASE0 is true and loaded on the rising edge of CLOCK with the contents of **M** register, which is the internal 16 bit data bus value.

Instruction Execute occurs during PHASE1, PHASE2, PHASE3.

{Control signals that make up instructions}
Control signals that aren't listed are negated (= 0).

ZERO = NOT(T15+T14+…+T0).
CARRY = T16.

Transfer Instructions

JMP

Jump to 10 bit address. {PSEL=1; PCE=1; PLOAD=1}

RET

Pop **R** stack to **P**. {PSEL=0; PCE=1; PLOAD=1; RCE=1; RPUSH=0}

JZ

Jump if **T** is zero. {PSEL=1; PCE=ZERO; PLOAD=ZERO}

JNC

Jump if CARRY is cleared. {PSEL=1; PCE=NOT(CARRY); PLOAD=NOT(CARRY)}

CALL

Push **P** to **R** stack. {RSEL=0; RCE=1; RPUSH=1}
Load **P** (bits 10..0) from **I**. {PSEL=1; PCE=1; PLOAD=1}

Memory Access Instructions

LDP

Load memory[**A**] to **T**.	{MSEL=1; TSEL1,0=11; TFUN1,0=01; TCE=1}
Push **T** to **N** stack.	{NPUSH=1; NCE=1}
Increment **A**.	{ACE=1}

LIT

Load in-line literal to **T**.	{MSEL=0; TSEL1,0=11; TFUN1,0=01; PCE=1; TCE=1}
Push **T** to **N** stack.	{NPUSH=1; NCE=1}

LD

Load memory[**A**] to **T**.	{MSEL=1; TSEL1,0=11; TFUN1,0=01; TCE=1}
Push **T** to **N** stack.	{NPUSH=1; NCE=1}

STP

Store **T** to memory[**A].**	{MSEL=1; D_OE=1}
Pop **N** stack to **T**.	{NPUSH=0; NCE=1; TCE=1; TSEL1,0=00; TFUN1,0=01}
Increment **A**.	{ACE=1}

ST

Store **T** to memory[**A**].	{MSEL=1; D_OE=1}
Pop **N** stack to **T**.	{NPUSH=0; NCE=1; TCE=1; TSEL1,0=00; TFUN1,0=01}

ALU Instructions

COM

Complement **T**.	{ALU1,0=00; TFUN1,0=00; TCE=1}

SHL

Shift **T** left 1 bit.	{TFUN1,0=10; TCE=1}

SHR

Shift **T** right 1 bit.	{TFUN1,0=11; TCE=1}

ADDC

if $\mathbf{T}_0$=1 then **N** + **T** => **T**.	{ALU1,0=11; TFUN1,0=00; TCE=($\mathbf{T}_0$=1)}
N preserved.	

XOR

Pop **N** stack.	{NPUSH=0; NCE=1}
N XOR **T** => **T**.	{ALU1,0=01; TFUN1,0=00; TCE=1}

AND

Pop **N** stack.	{NPUSH=0; NCE=1}
N AND **T** => **T**.	{ALU1,0=10; TFUN1,0=00; TCE=1}

ADD

Pop **N** stack.	{NPUSH=0; NCE=1}
N + **T** => **T**.	{ALU1,0=11; TFUN1,0=00; TCE=1}

Register Instructions

POP

Push **T** to **N** stack.	{NCE=1; NPUSH=1}
Pop **R** stack to **T.**	{RCE=1; RPUSH=0; TSEL1,0=10; TCE=1; TFUN1,0=01}

LDA

Push **T** to **N** stack.	{NCE=1; NPUSH=1}
Load **A** to **T**.	{TSEL1,0=01; TCE=1; TFUN1,0=01}

DUP

Push **T** to **N** stack.	{NCE=1; NPUSH=1}

PUSH

Push **T** to **R** stack.	{RCE=1; RPUSH=1}
Pop **N** stack to **T**.	{NCE=1; NPUSH=0; TSEL1,0=00; TCE=1; TFUN1,0=01}

STA

Store **T** to **A**.	{ACE=1; ALOAD=1}
Pop **N** stack to **T**.	{NCE=1; NPUSH=0; TSEL1,0=00; TCE=1; TFUN1,0=01}

NOP

No operation.	{ }

DROP

Pop **N** stack to **T**.	{NCE=1; NPUSH=0; TSEL1,0=00; TCE=1; TFUN1,0=01}