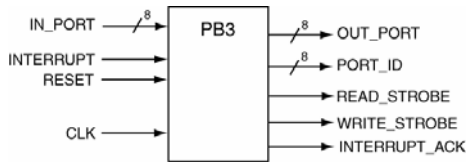
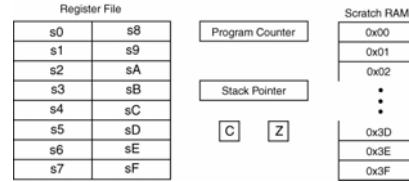


# PicoBlaze3 Overview Sheet

© Copyright: 2007 bryan mealy



schematic diagram



programming model

## PicoBlaze Instruction Set:

Program Control			Interrupt	Input/Output
JUMP label	CALL label	RETURN	RETURNI ENABLE	INPUT sX,pp
JUMP Z, label	CALL Z,sr_label	RETURN Z	RETURNI DISABLE	INPUT sX,(sY)
JUMP NZ, label	CALL NZ,sr_label	RETURN NZ	ENABLE INTERRUPT	OUTPUT sX,pp
JUMP C, label	CALL C,sr_label	RETURN C	DISABLE INTERRUPT	OUTPUT sX,(sY)
JUMP NC, label	CALL NC,sr_label	RETURN NC		

Logical		Arithmetic		Shift & Rotate		Storage
LOAD sX,kk	LOAD sX,sY	ADD sX,kk	ADD sX,sY	SR0 sX	SR0 sX	STORE sX,s
AND sX,kk	AND sX,sY	ADDCY sX,kk	ADDCY sX,sY	SR1 sX	SR1 sX	STORE sX,(sY)
OR sX,kk	OR sX,sY	SUB sX,kk	SUB sX,sY	SRX sX	SRX sX	FETCH sX,ss
XOR sX,kk	XOR sX,sY	SUBCY sX,kk	SUBCY sX,sY	SRA sX	SRA sX	FETCH sX,(sY)
TEST sX,kk	TEST sX,sY	COMPARE sX,kk	COMPARE sX,sY	RR sX	RR sX	STORE sX,ss

### Fun Facts:

- Maximum program size: 1024 instructions
- 16 general purpose registers (GPRs)
- Stack: 16 deep (subroutine & interrupt nesting level)

### I/O:

- Port Mapped device (8-bit port addresses)
- 8-bit GPR-based Input and Output

### Interrupt Architecture:

- Interrupt on: 1) interrupt input high voltage, 2) interrupt enabled
- One maskable external interrupt (see interrupt group)
- Vector interrupt: vector address 0x3FF
- Context saving: C & Z flags saved (restored on RETURNI instructions)
- Interrupt automatically disabled on interrupt

## Bit Masking:

bit setting: OR with '1'	bit clearing: AND with '0'	bit toggling: XOR with '1'
OR s3,01 ;set bit 1	AND s3,FE ;clear bit 1: ;s3 is altered TEST s3,FE ;test bit 1 ;s3 is not altered	XOR s3,01 ;toggle bit 1

## Conditional JUMPs Constructs:

if/else construct	<pre> main:      INPUT    s0,IN_PORT    ; grab data            AND      s0,s0_      ; nop with flag setting            JUMP    NZ,not_zero  ; jump if s0 is non-zero            LOAD    s1,00        ; clear bits in s1            JUMP    out_val      ; jump to output instruction            ; not_zero:  LOAD    s1,FF        ; set bits in s1 out_val:  OUTPUT   s1,OUT_PORT  ; output some data            JUMP    main         ; do it again         </pre>
iterative construct	<pre> loop:      LOAD    s3,08        ; load iterative count value            ;            ; so something meaningful            SUB     s3,01        ; decrement iteration variable            JUMP    NZ,loop      ; do it again if count non-zero            ; loop_done: ; do something else..         </pre>
conditional construct	<pre> loop:      LOAD    s0,00        ; clear register (accumulator)            INPUT   s1,IN_PORT  ; grab some data            ADD     s0,s1_      ; add value to s0            JUMP    NC,loop      ; repeat if no carry            ; loop_done: ; do something else..         </pre>

## Equality/Inequality:

- C and Z flag are used to establish relationship between register as follows:

Operation:		
		SUB      sA, sB
		COMPARE sA, sB
C	Z	Comment
0	0	sA > sB
1	0	sA < sB