

# 1 The Pulse Programmer

## 1.1 Overview

## 1.2 Registers

The state machine has the following registers:

W	Main work register
W1	auxilliary work register
PC	code address
INDF	indirect memory address
shutter_mask	mask for shutter values
shutter_reg	shutter value buffer
counter_reg	counter gate buffer

## 1.3 Commands

**NOP** No operation

**DDSFRQ channel**, *variable* write frequency (32 most significant bits) from variable to DDS channel

**DDSFRQFINE channel**, *variable* write frequency (16 least significant bits) from variable to DDS channel

**DDSAMP channel**, *variable* write amplitude from variable to DDS channel

**DDSPHS channel**, *variable* write phase from variable to DDS channel

**DDSCHN**

**SHUTTER**

**COUNT**

**COUNT1**

**COUNTBOTH**

**DELAY**

**LDWR** *variable* load value from variable into W register

**LDWR1**

**LDWI** load value from the address pointed to by INDF into W register

**STWR** *variable* store value in W register into variable

**STWR1**

**STWI** store value from W register into address pointed to by INDF

**LDINDF** *variable* load the contents of *variable* into the INDF register

**ANDW** *variable*  $W = W \& \text{variable}$

**ADDW** *variable*  $W = W + \text{variable}$

**INC** *variable*  $W = \text{variable} + 1$

**DEC** *variable*  $W = \text{variable} - 1$

**CLRW**  $W = 0$

**CLRW1**  $W1 = 0$

**CMP** *variable* Set W to 0 if  $W \neq \text{variable}$

**CMP1**

**JMP** *label* Jump to *label*

**JMPZ** *label* Jump to *label* if  $W = 0$

**JMPZ1** *label* Jump to *label* if  $W1 = 0$

**JMPNZ** *label* Jump to *label* if  $W \neq 0$

**JMPNZ1** *label* Jump to *label* if  $W1 \neq 0$

**SHUTTERMASK** *variable* Set internal register shutter\_mask to *variable*

**ASYNCSHUTTER** *variable* Update internal shutter register, bits set in shutter\_mask are updated with the bits from *variable*

**COUTERMASK** *variable* Set the internal register with gate signals for the 8 counters and timestampers. Bits 7:0 gate counters 7:0, bits 15:8 gate timestamping on channels 7:0.

**TRIGGER** *variable* Set internal trigger register

**UPDATE** *variable* Update shutters, counter gates, triggers and start the delay counter with the value in *variable*

**WAIT** wait until the delay counter expires

**LDCOUNT** *counterchannel* load the last counter value from *counterchannel* into W register

**WRITEPIPE** write the value in W into the pipe to the host computer

**READPIPE** read a value from the pipe from the host computer into the W register. If there is no new data in the pipe, the last value in the pipe is used.

**LDTDCCOUNT** load the value from the global tdc counter into W

**CMPEQUAL** *variable* compare W and *variable* and set the internal compare bit to true if  $W=variable$

**JMPCMP** *label* Jump to label if the internal compare bit is set

**JMPNCMP** *label* Jump to label if the internal compare bit is not set

**JMPPIPEAVAIL** *label* Jump to label if the pipe from the host computer has data

**JMPPIPEEMPTY** *label* Jump to label if the pipe from the host computer is empty

**READPIPEINDF** Read the value from the pipe from the host computer in the INDF register

**WRITEPIPEINDF** Write the value from the INDF register into the pipe to the host computer

**STOP** Stop execution