



# USB-DSP

## FEATURES

- 100% Instructions run in just one single clock.
- Wide set of I/O, Algebraic and Boolean instructions.
- Specific set of instructions for the generation and recognition of sequences.
- Operating frequency 1.5 to 50MHz.
- Low consumption.
- High efficiency in generation of waveforms at the maximum frequency.
- 64-bit I/O controlled by DSP.
- 16-bit I/O under direct USB control.
- External 16-bit bus (16 Address and 16 Data) for the peripherals management.
- 8 16-bit registers for generic use.
- 128 word RAM data.
- 3584 program steps.
- USB 2.0 Connection with 66 Mbyte/s bandpass.
- Totally transparent use of the USB, it is just a simple register for the processor.
- Algebraic type assembler language having many similarities to the BASIC language.
- Integrated development environment including Editor, Assembler, Debugger and waveform viewer.
- DLL Library and assembler for the management from third (C/C++ LabView etc.).
- USB self-powered.
- Low price.
- Small size (58x85x10mm).
- Only chip available.
- Addition of new instructions and internal peripherals on demand.

## DESCRIPTION

This is the ideal device if you want to create complex and fast patterns and yet quickly acquire large quantities of data.

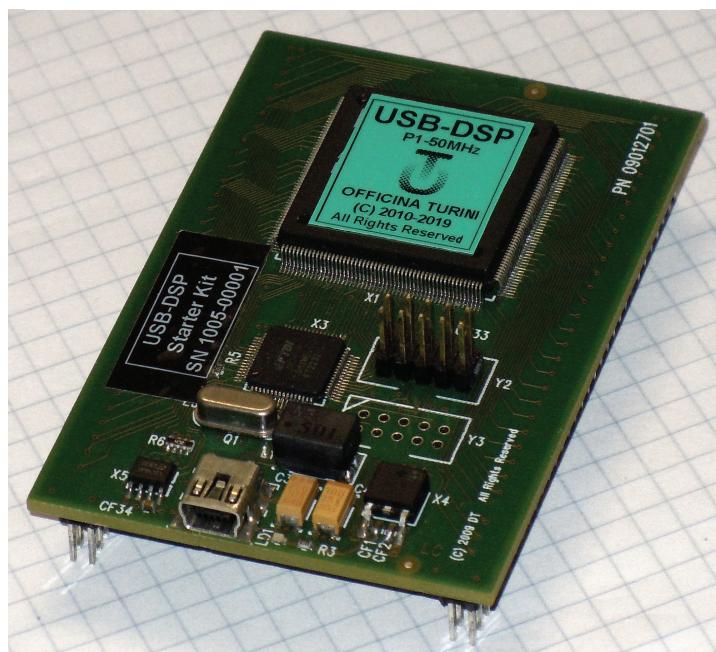
From your personal computer you can create complex real-time acquisition functions but with the ease of BASIC programming.

The created code is loaded and run through the same USB.

The functions of programming, debugging, execution and verification of timing take place within an IDE (MDI) which includes an editor with Syntax Highlighting.

## APPLICATIONS

- Frame grabber
- Diagnostics
- Real Time Interface
- Smart Port
- Sequencer
- Pulse Generator



REV. A3 2017/10/12

Neither the whole nor any part of the information contained in, or the product described in this datasheet, may be adapted or reproduced in any material or electronic form without the prior written consent of the copyright holder. This product and its documentation are supplied on an as-is basis and no warranty as to their suitability for any particular purpose is either made or implied. O.T. will not accept any claim for damages howsoever arising as a result of use or failure of this product. Your statutory rights are not affected. This product or any variant of it is not intended for use in any medical appliance, device or system in which the failure of the product might reasonably be expected to result in personal injury. This document provides preliminary information that may be subject to change without notice. No freedom to use patents or other intellectual property rights is implied by the publication of this document.

Via G. Cei 67, 56021 Cascina, Pisa Italy

Tel: +039-050-8665471

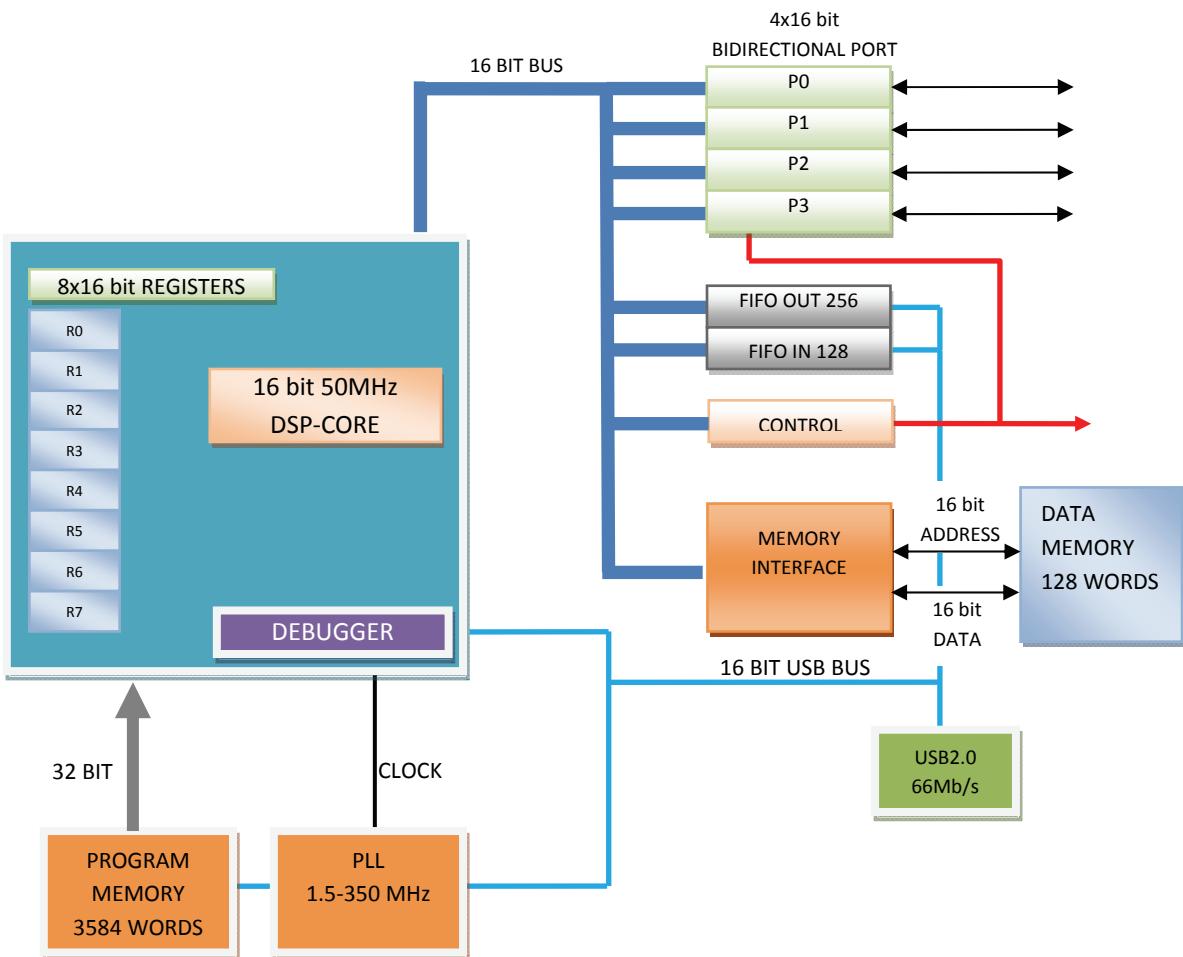
[www.officinaturini.com](http://www.officinaturini.com)

Fax: +039-050-8665471

© O.T., 2010-2019

## ARCHITECTURE

USB-DSP is organized in a Harward architecture. It has 8 16-bit generic registers a and a ALU register. Moreover, it has a 128 word data RAM.



All the blocks here shown are in the inner of the chip, but the USB, where a FTDI 2232H device has been used.

## INTRODUCTION

The I/O of the processor is realized on 4 16-bits directional ports that you can organize in input/output by groups of 8-bits.

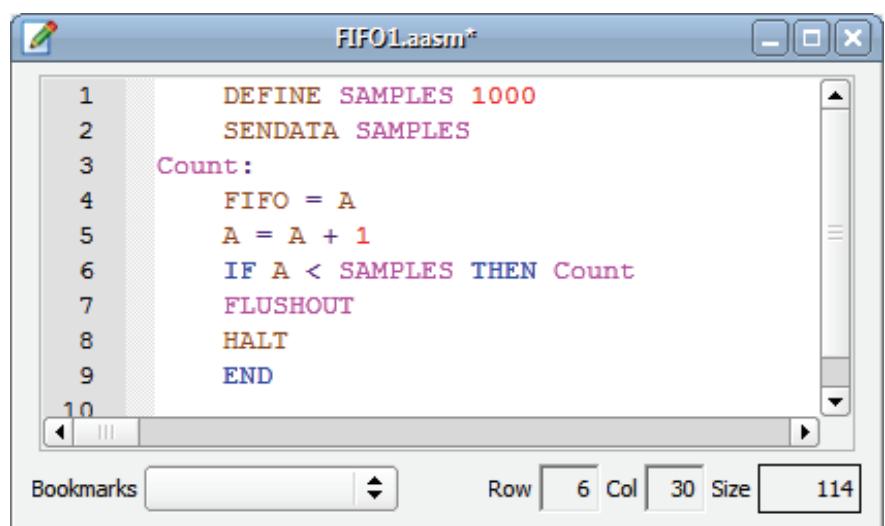
Other than the typical mathematical and logic operations, the instructions also include specific instructions for the timing generation and the recognition of sequences.

The operating frequency of the processor can be set on a continuous 1.5 to 50 MHz by the internal PLL.

Let's examine a simple example to show the philosophy of the system.

In this example, the program executes a count on the ALU register and writes the content in the FIFO memory of USB. When the FIFO (256 words) reaches 128 words, it automatically run a transfer from to PC by USB.

Let analyze row by row what the code is doing:



The screenshot shows a software window titled "FIFO1.asm\*". The code is as follows:

```
1      DEFINE SAMPLES 1000
2      SENDATA SAMPLES
3      Count:
4          FIFO = A
5          A = A + 1
6          IF A < SAMPLES THEN Count
7          FLUSHOUT
8          HALT
9
10
```

Below the code, there are buttons for "Bookmarks", "Row" (set to 6), "Col" (set to 30), "Size" (set to 114), and a scroll bar.

1. Define the name SAMPLES which is associated with the numerical value of 1000.
2. The command SENDATA notify the PC that SAMPLES (1000) words will be transferred.
3. Definition of a program position by the “Count” label.
4. Write the content of A in the FIFO (take into account all the registers are at 0 after the reset).
5. Increase by 1 the content of A.
6. Check the count of A reaches the SAMPLES value.
7. Once the maximum value has been reached, it runs a request to empty the FIFO because this command is automatically run only when almost 128 words has been written.
8. The processor is stopped by the HALT instruction.
9. Statement indicating the end of the program.

The program is made by a total of 6 instructions, they are run 1000 times, so it needs a total time of 120  $\mu$ s with a 50 MHz frequency of clock.

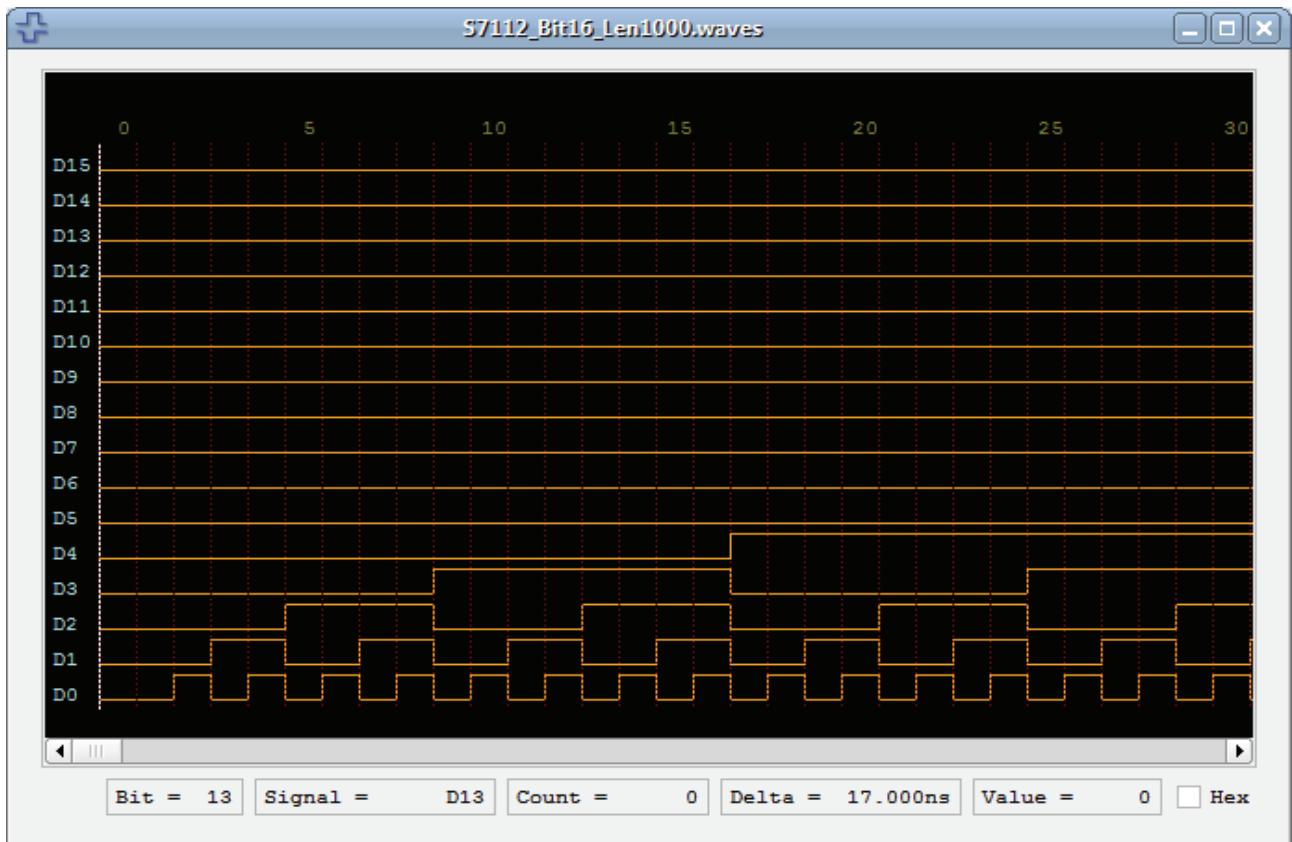
It is legitimate to ask why that 6 instructions when those involved in the loop are only 3, just look at the listing below.

```

1 [00011]-----:----- DEFINE SAMPLES 1000
2 [00012]-----:----- SENDDATA SAMPLES
3 [00019]-----:----- END
4 [00014]0x00000000:----- [Count]
5 [00014]0x00000000:0x54800000 FIFO = A
6 [00015]0x00000001:0x11E80001 A = A + 1
7 [00016]0x00000002:0x1DE803E8 IF A < SAMPLES THEN Count
8 [00016]0x00000003:0x20E20000 JL Count
9 [00016]0x00000004:0x00000000 NOP
10 [00016]0x00000005:0x00000000 NOP
11 [00017]0x00000006:0x41000000 FLUSHOUT
12 [00018]0x00000007:0x60000000 HALT
13 [00018]0x00000008:0x00000000 NOP
14 [00018]0x00000009:0x00000000 NOP

```

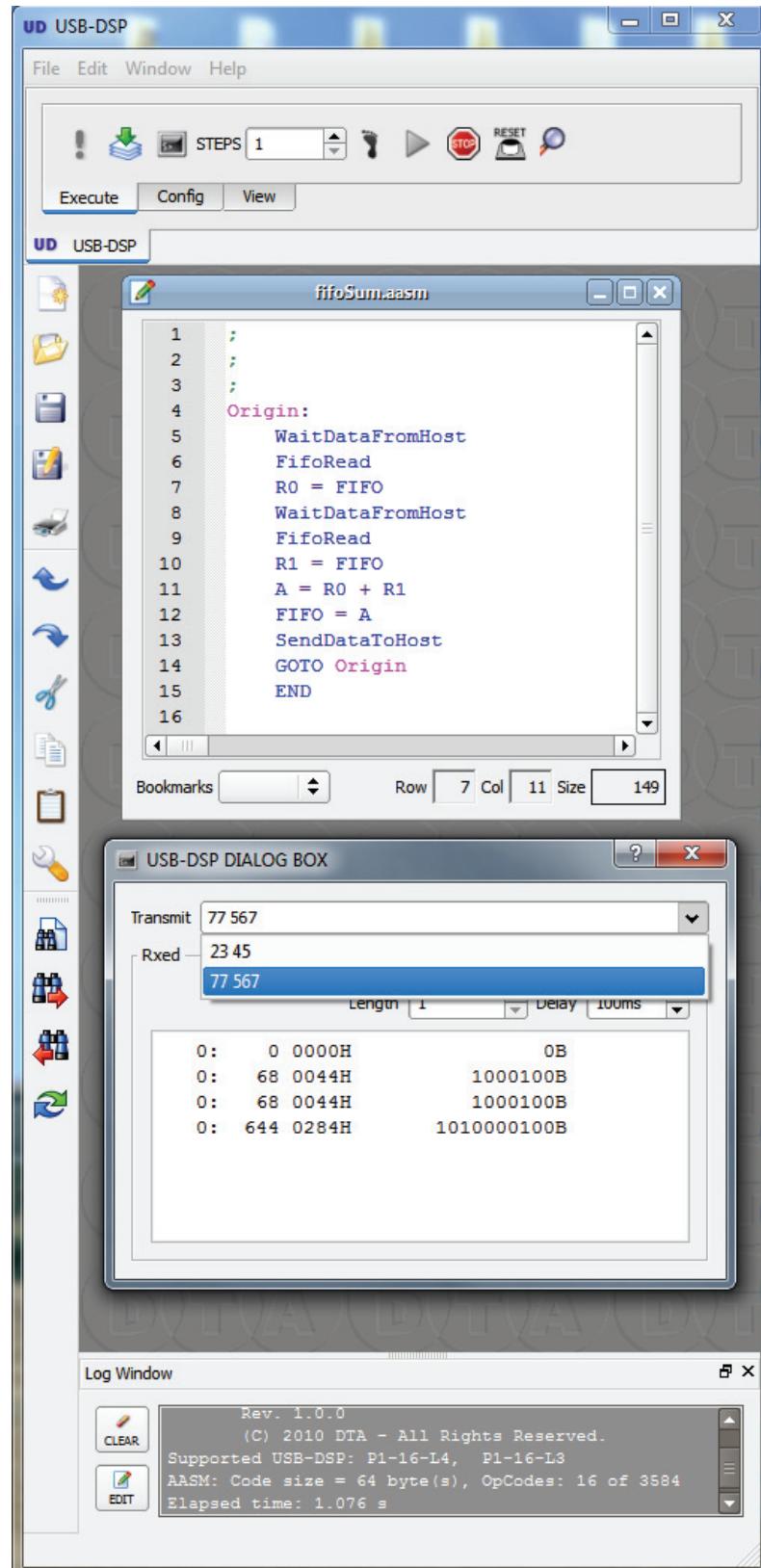
The IF produces an instruction CMP (Compare) at the line 6, after which there is a conditional jump and the underlying NOP, necessary to avoid the execution of the instruction at the line 7 for the processor pipeline. Asking for the display of the data received from the PC, we'll have:



This is a way to display the data received from the processor or to monitor the activity of one of the 4 16-bits ports.

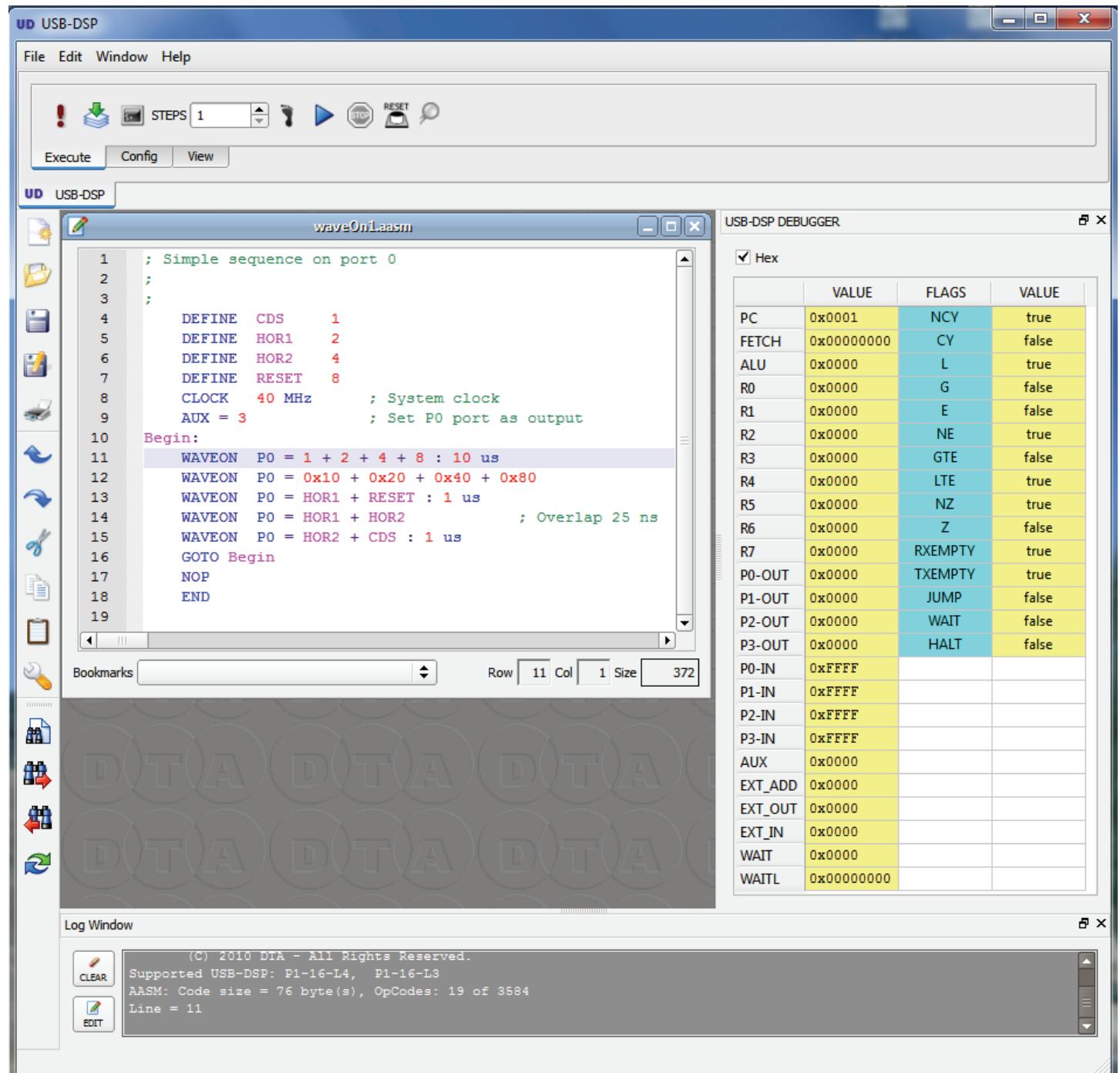
But it can also talk directly to the processor by a simple console.

In the example below, the written program executes the sum of two numbers received from the PC by the dialogue window and then it transmits the result.



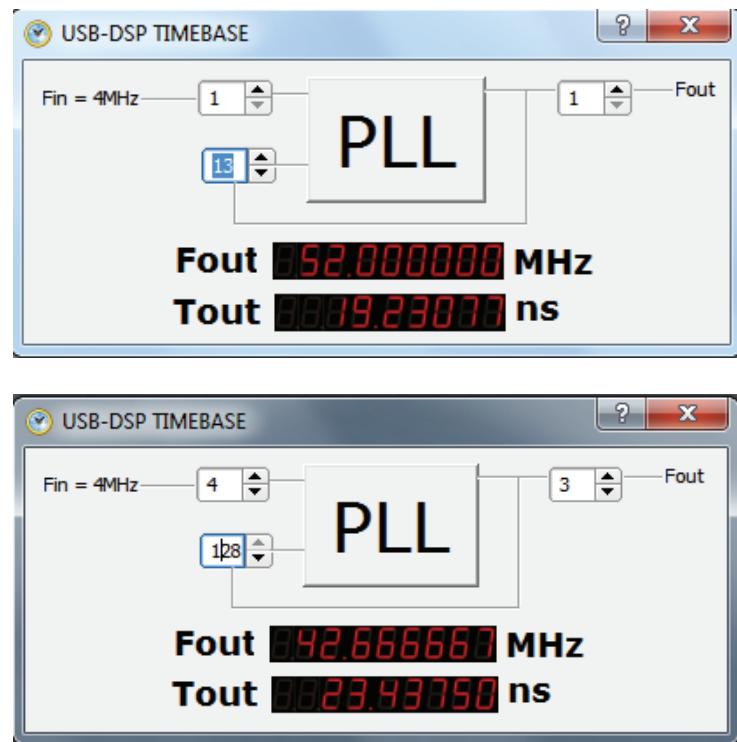
The integrated system for writing programs and their execution also includes a DEBUG mode for the step-step tracking.

All the registers of the processor are shown and the progress of the program is visible on the editor:



All the functions offered by the integrated environment (IDE) are also executable by the library (DLL) present in the SDK.

The operating frequency of the processor can be set from 1.5 to 50 MHz by the proper control.



USB-DSP is guaranteed for a operating frequency of 50MHz, but actually it has a minimum operating frequency of 54 MHz to a maximum of 60 MHz.

In the future, we'll also be available the -2 version that will offer guaranteed performances at 66 MHz.

## PERFORMANCES

The performances it can achieve are very naturally related to the quality of the program written, for that reason it is important to know very well the instructions that the core (P1) of DSP-USB provides.

At 50 MHz clock every instruction "costs" 20 ns, so it's a good thing to take this into account.

Also for achieving slower timing, specific instructions delay (WAIT) or instructions for complex synchronization (MATCH) are available.

With a single instruction can be executed 24-bit delays or recognize 16-bit sequences.

In the example below you can see a simple program that generates a square wave at the possible maximum speed on port P0 and in the following image there is the feedback about what physically happens on bit 0 of port P0.

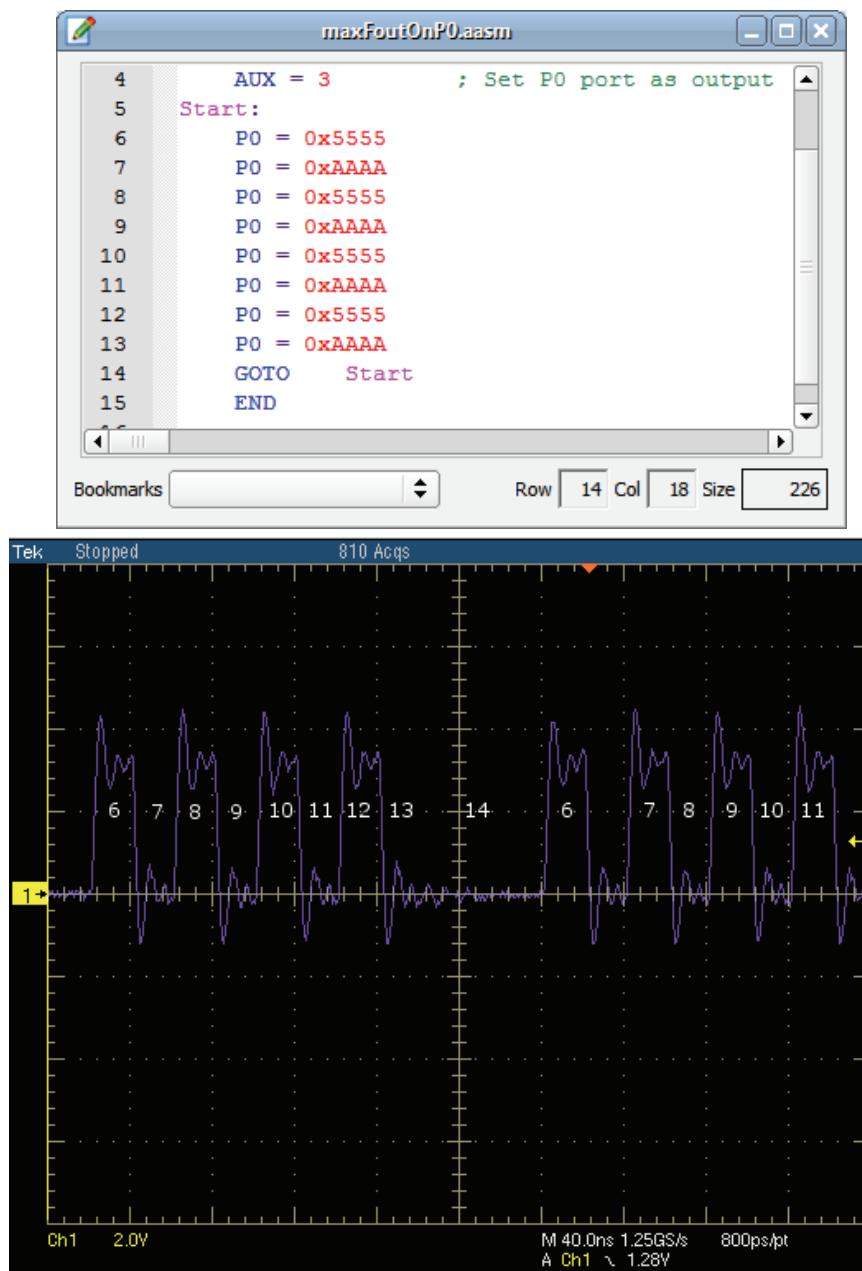


Image taken by the oscilloscope showing the activity of the bit 0 of the port P0, the numbers written on the waveform are the line numbers of the program.

## APPLICATIONS

We said that USB-DSP is strongly indicated where necessary to create waveforms very precise and acquire large amounts of data at high speed. In the example below the listed program runs a CCD sensor, creating necessary vertical and horizontal steps; in addition, it reads and transmits to the PC the data from the A/D converter.

Use of the ports:

P0 - Sensor phases and A/D synchronization

P1 - Data from the Converter (16-bit)

P2 – Trigger input

Let analyze it row by row:

- 1.
- 2.
- 3.
- 4.
- 5.
- 6.
- 7.
8. The first 8 rows use the command LEGEND that define the use of a bit on the specified port. It could also write at line 1:  
DEFINE HOR1 1  
LEGEND, other than to create the same type of definition, also reports a documentation into the binary created useful for the data analysis as we'll see later.
9. We found the CLOCK instruction: it indicates the operating frequency of the processor to the assembler and it will be useful to create the timing instruction.
10. Line of comment
11. Set the port P0 in output
12. Set the bit 1 (of a generic port) active and without inversion
13. Definition of a label indicating the beginning of the main cycle
14. Use the register R0 as row counter (512)
15. Instruction stopping the processor until the trigger condition is satisfied P2[1] = 1
16. Label identifying the beginning of the vertical transfer
17. Set the signal HOR2 keeping it active for 500 ns
18. Set the signals VER2 and HOR2 keeping them active for per 5  $\mu$ s
19. Set the signals VER1 and HOR2 keeping them active for per 5  $\mu$ s
20. Set the signals VER2 and HOR2 keeping them active for per 5  $\mu$ s

```

1      LEGEND  P0,0,HOR1
2      LEGEND  P0,1,HOR2
3      LEGEND  P0,2,RESET
4      LEGEND  P0,3,CLAMP
5      LEGEND  P0,4,SAMPLE
6      LEGEND  P0,5,ADTRG
7      LEGEND  P0,6,VER1
8      LEGEND  P0,7,VER2
9      CLOCK   50 MHz
10     ;
11     AUX = 3
12     MATCHSET 1,0
13     Begin:
14     R0 = 512
15     MATCH      P2,0
16     VShift:
17     WAVEON  P0 = HOR2 : 500 ns
18     WAVEON  P0 = VER2 + HOR2 : 5 us
19     WAVEON  P0 = VER1 + HOR2 : 5 us
20     WAVEON  P0 = VER2 + HOR2 : 5 us
21     WAVEON  P0 = HOR2 : 500 ns
22     R1 = 768
23     GetLine:
24     WAVEON  P0 = HOR2 + RESET
25     WAVEON  P0 = HOR2 + CLAMP
26     WAVEON  P0 = HOR1 + SAMPLE
27     WAVEON  P0 = HOR1 + ADTRG
28     FIFO = P1
29     R1 = R1 - 1
30     JUMP NZ, GetLine
31     NOP
32     NOP
33     R0 = R0 - 1
34     JUMP NZ, VShift
35     NOP
36     NOP
37     FLUSHOUT
38     MATCH      P2,1
39     GOTO Begin
40
41

```

21. Set the signals HOR2 keeping it active for 500 ns
22. Use the register R1 as column counter (768)
23. Define the label Getline
24. Set the signals RESET and HOR2 keeping them active for a clock cycle (20ns)
25. Set the signals CLAMP and HOR2 keeping them active for a clock cycle (20ns)
26. Set the signals SAMPLE and HOR1 keeping them active for a clock cycle (20ns)
27. Set the signals ADTRG and HOR1 keeping them active for a clock cycle (20ns)
28. Send the content of the A/D to the FIFO
29. Decrease the content of the register R1 used as horizontal pixels counter
30. Read another line if R1 hasn't reached the 0
31. Instruction NOP for the pipeline
32. Instruction NOP for the pipeline
33. Decrease the content of the register R0 used as vertical line counter
34. Read another line if R0 hasn't reached the 0
35. Instruction NOP for the pipeline
36. Instruction NOP for the pipeline
37. When the program arrives at this point, a complete frame has been sent and with this instruction it asks for the total emptying of the FIFO
38. Wait until the trigger signal is inactive again
39. Start again from the beginning for the next frame

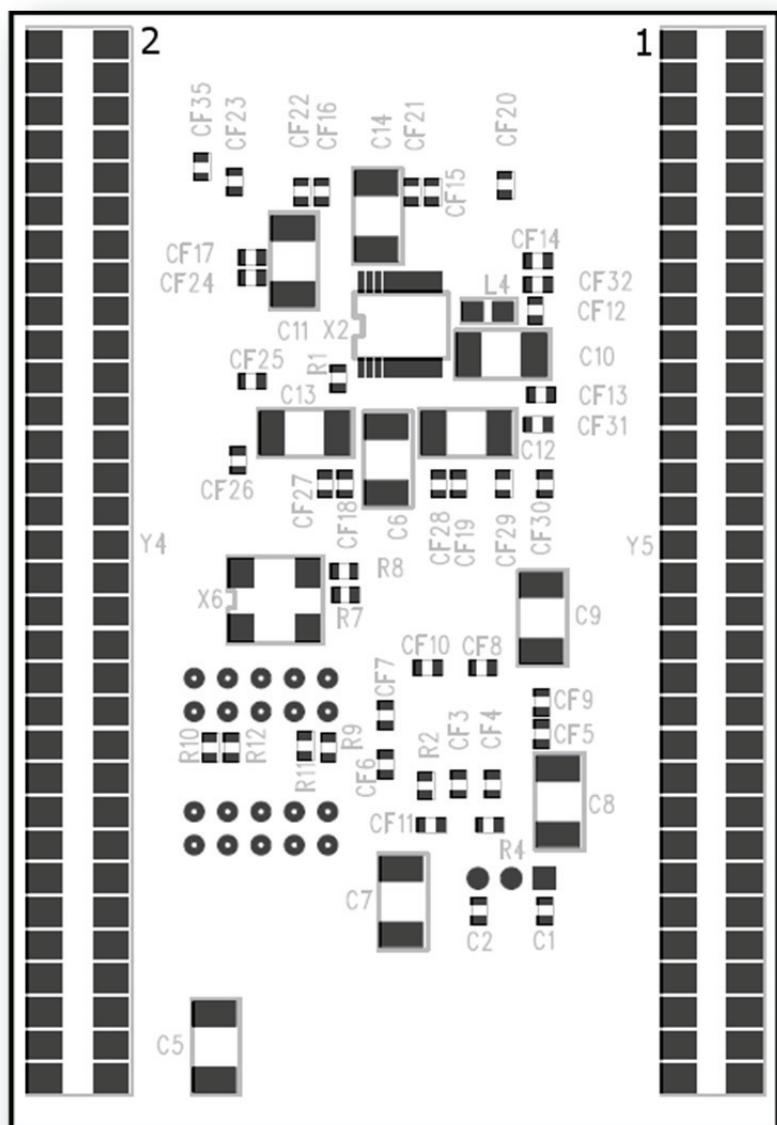
This example is a simplified version of the program as the reading of a CCD sensor results in some operations to be carried out prior to the actual acquisition, but could not be described as an example something too complicated. The final performances are a sampling of the CCD sensor at a frame rate of 5Mpix/s (10 MB/s bandwidth used on USB) so that a frame is read in about 16 ms (over 60 frames per second) considering the time for the vertical transfer and the reading of all the pixels.

## I/O CONNECTOR

For I/O operations, the USB-DSP uses two 64 pin 100 mils (Y4-Y5) connectors with 800 mils distance, so you can easily use them it on prototype boards with 100 mils steps.

On the connectors there are the four system ports (P0-3), the 16-bit external bus for the management of the optional peripherals and various control and powering.

All the signals are LVTTL compatible.



Disposition of the connectors in the bottom side of the board

Y4										
DESCRIPTION	IO	USB-DSP	SCHEMATIC SIGNALS	PIN	PIN	SCHEMATIC SIGNALS	USB-DSP	IO	DESCRIPTION	
500 mA max		VCC	+3,3V	1	2	+3,3V	VCC		500 mA max	
P0 port	IO	P0[1]	IO061	3	4	IO060	P0[0]	IO	P0 port	
P0 port	IO	P0[3]	IO063	5	6	IO062	P0[2]	IO	P0 port	
P0 port	IO	P0[5]	IO065	7	8	IO064	P0[4]	IO	P0 port	
P0 port	IO	P0[7]	IO067	9	10	IO066	P0[6]	IO	P0 port	
P0 port	IO	P0[9]	IO069	11	12	IO068	P0[8]	IO	P0 port	
P0 port	IO	P0[11]	IO071	13	14	IO070	P0[10]	IO	P0 port	
P0 port	IO	P0[13]	IO073	15	16	IO072	P0[12]	IO	P0 port	
P0 port	IO	P0[15]	IO075	17	18	IO074	P0[14]	IO	P0 port	
P1 port	IO	P1[1]	IO077	19	20	IO076	P1[0]	IO	P1 port	
P1 port	IO	P1[3]	IO079	21	22	IO078	P1[2]	IO	P1 port	
P1 port	IO	P1[5]	IO081	23	24	IO080	P1[4]	IO	P1 port	
P1 port	IO	P1[7]	IO083	25	26	IO082	P1[6]	IO	P1 port	
P1 port	IO	P1[9]	IO085	27	28	IO084	P1[8]	IO	P1 port	
P1 port	IO	P1[11]	IO087	29	30	IO086	P1[10]	IO	P1 port	
P1 port	IO	P1[13]	IO089	31	32	IO088	P1[12]	IO	P1 port	
P1 port	IO	P1[15]	IO091	33	34	IO090	P1[14]	IO	P1 port	
P2 port	IO	P2[1]	IO093	35	36	IO092	P2[0]	IO	P2 port	
P2 port	IO	P2[3]	IO095	37	38	IO094	P2[2]	IO	P2 port	
P2 port	IO	P2[5]	IO097	39	40	IO096	P2[4]	IO	P2 port	
P2 port	IO	P2[7]	IO099	41	42	IO098	P2[6]	IO	P2 port	
P2 port	IO	P2[9]	IO101	43	44	IO100	P2[8]	IO	P2 port	
P2 port	IO	P2[11]	IO103	45	46	IO102	P2[10]	IO	P2 port	
P2 port	IO	P2[13]	IO105	47	48	IO104	P2[12]	IO	P2 port	
P2 port	IO	P2[15]	IO107	49	50	IO106	P2[14]	IO	P2 port	
P3 port	IO	P3[1]	IO109	51	52	IO108	P3[0]	IO	P3 port	
P3 port	IO	P3[3]	IO111	53	54	IO110	P3[2]	IO	P3 port	
P3 port	IO	P3[5]	IO113	55	56	IO112	P3[4]	IO	P3 port	
P3 port	IO	P3[7]	IO115	57	58	IO114	P3[6]	IO	P3 port	
	-	IO117	59	60	IO116	-				
4MHz CLOCK	O	MCLK	MCLK	61	62	/RES	/RESET	O	RESET (Active low)	
		GND	GND	63	64	GND	GND			

**Y5**

DESCRIPTION	IO	USB-DSP	SCHEMATIC SIGNALS	PIN	PIN	SCHEMATIC SIGNALS	USB-DSP	IO	DESCRIPTION
500 mA max		VCC	+3,3V	1	2	+3,3V	VCC		500 mA max
P3 port	IO	P3[9]	IO058	3	4	IO059	P3[8]	IO	P3 port
P3 port	IO	P3[11]	IO056	5	6	IO057	P3[10]	IO	P3 port
P3 port	IO	P3[13]	IO054	7	8	IO055	P3[12]	IO	P3 port
P3 port	IO	P3[15]	IO052	9	10	IO053	P3[14]	IO	P3 port
	-	IO050	11	12	IO051	-			
	-	IO048	13	14	IO049	-			
	-	IO046	15	16	IO047	-			
	-	IO044	17	18	IO045	-			
P1 RESET	O	RESOUT	IO042	19	20	IO043	-		
P1 HALT	O	HALT	IO040	21	22	IO041	READY	I	Ready signal (Internal pullup)
IO Address	O	A[0]	IO038	23	24	IO039	CLKOUT	O	P1 CLOCK
IO Address	O	A[2]	IO036	25	26	IO037	A[1]	O	IO Address
IO Address	O	A[4]	IO034	27	28	IO035	A[3]	O	IO Address
IO Address	O	A[6]	IO032	29	30	IO033	A[5]	O	IO Address
IO Address	O	A[8]	IO030	31	32	IO031	A[7]	O	IO Address
IO Address	O	A[10]	IO028	33	34	IO029	A[9]	O	IO Address
IO Address	O	A[12]	IO026	35	36	IO027	A[11]	O	IO Address
IO Address	O	A[14]	IO024	37	38	IO025	A[13]	O	IO Address
IO Data	IO	D[0]	IO022	39	40	IO023	A[15]	IO	IO Data
IO Data	IO	D[2]	IO020	41	42	IO021	D[1]	IO	IO Data
IO Data	IO	D[4]	IO018	43	44	IO019	D[3]	IO	IO Data
IO Data	IO	D[6]	IO016	45	46	IO017	D[5]	IO	IO Data
IO Data	IO	D[8]	IO014	47	48	IO015	D[7]	IO	IO Data
IO Data	IO	D[10]	IO012	49	50	IO013	D[9]	IO	IO Data
IO Data	IO	D[12]	IO010	51	52	IO011	D[11]	IO	IO Data
IO Data	IO	D[14]	IO008	53	54	IO009	D[13]	IO	IO Data
IO Read (Active low)	O	/IORD	IO006	55	56	IO007	D[15]	IO	IO Data
	-	IO004	57	58	IO005	/IOWR	O	IO Write (Active low)	
		DBG2	IO002	59	60	IO003	-		
		DBG0	IO000	61	62	IO001	DBG1		
		GND	GND	63	64	GND	GND		

# USB-DSP DC and Switching Characteristics

## Operating Conditions

Stresses beyond those listed in [Table 2-1](#) may cause permanent damage to the device.

Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Absolute Maximum Ratings are stress ratings only; functional operation of the device at these or any other conditions beyond those listed under the Recommended Operating Conditions specified in [Table 2-2 on page 2-2](#) is not implied.

**Table 2-1 • Absolute Maximum Ratings**

Symbol	Parameter	Limits	Units
VCC	DC core supply voltage	-0.3 to 1.65	V
V <sub>JTAG</sub>	JTAG DC voltage	-0.3 to 3.75	V
V <sub>PUMP</sub>	Programming voltage	-0.3 to 3.75	V
V <sub>CCPLL</sub>	Analog power supply (PLL)	-0.3 to 1.65	V
V <sub>CCI</sub>	DC I/O output buffer supply voltage	-0.3 to 3.75	V
V <sub>VMV</sub>	DC I/O input buffer supply voltage	-0.3 to 3.75	V
V <sub>I</sub>	I/O input voltage	-0.3 V to 3.6	V
T <sub>TG</sub> <sup>2</sup>	Storage temperature	-65 to +150	°C
T <sub>J</sub> <sup>2</sup>	Junction temperature	+125	°C

*Notes:*

1. The device should be operated within the limits specified by the datasheet. During transitions, the input signal may undershoot or overshoot according to the limits shown in [Table 2-4 on page 2-3](#).

**Table 2-2 • Recommended Operating Conditions**<sup>1,2</sup>

Symbol	Parameter	Commercial	Industrial	Units
T <sub>A</sub>	Ambient temperature	0 to +70	-40 to +85	°C
T <sub>J</sub>	Junction temperature	0 to 85	-40 to 100	
V <sub>CC</sub> <sup>2</sup>	1.5 V DC core supply voltage	1.425 to 1.575	1.425 to 1.575	V
V <sub>JTAG</sub>	JTAG DC voltage	1.4 to 3.6	1.4 to 3.6	V
V <sub>PUMP</sub>	Programming voltage	3.0 to 3.6	3.0 to 3.6	V
	Operation <sup>3</sup>	0 to 3.6	0 to 3.6	V
V <sub>CCPLL</sub>	Analog power supply (PLL)	1.4 to 1.6	1.4 to 1.6	V
V <sub>CCI</sub> and V <sub>VMV</sub>	1.5 V DC supply voltage	1.425 to 1.575	1.425 to 1.575	V
	1.8 V DC supply voltage	1.7 to 1.9	1.7 to 1.9	V
	2.5 V DC supply voltage	2.3 to 2.7	2.3 to 2.7	V
	3.3 V DC supply voltage	2.7 to 3.6	2.7 to 3.6	V
	3.3 V wide range DC supply voltage <sup>4</sup>	3.0 to 3.6	3.0 to 3.6	V
	LVDS/B-LVDS/M-LVDS differential I/O	2.375 to 2.625	2.375 to 2.625	V
	LVPECL differential I/O	3.0 to 3.6	3.0 to 3.6	V

*Notes:*

1. All parameters representing voltages are measured with respect to GND unless otherwise specified.
2. The ranges given here are for power supplies only.
3. VPUMP can be left floating during operation (not programming mode).
4. 3.3 V wide range is compliant to the JDEC8b specification and supports 3.0 V V<sub>CCI</sub> operation.

**Table 2-3 • Flash Programming Limits – Retention, Storage and Operating Temperature<sup>1</sup>**

Product Grade	Program Retention (biased/unbiased)	Maximum Storage Temperature TSTG (°C) 2	Maximum Operating Junction Temperature TJ (°C) 2
Commercial	20 years	110	100
Industrial	20 years	110	100

*Notes:*

1. This is a stress rating only; functional operation at any condition other than those indicated is not implied.
2. These limits apply for program/data retention only. Refer to [Table 2-2](#) for device operating conditions and absolute limits.

## CONTACTS

**Officina Turini di Manuele Turini**  
Via G. Cei 67  
56021 Cascina  
Pisa  
ITALY  
Phone: 390508665471  
FAX: 390508665471  
Email (Support): [info@officinaturini.com](mailto:info@officinaturini.com)  
WEB: [www.officinaturini.com](http://www.officinaturini.com)