Features

- 8 Bit RISC Processor
  - Atmel AVR® ISA compatible
  - 1K Byte Instruction RAM
  - 128 Byte Flash BIOS (JTAG writeable)
  - 32 Registers
  - 4 Level hardware stack
  - Up to 25MIPS speed
- Clock manager
  - 50MHz Crystal Oscillator
  - NCO for system clock
  - Dynamic PLL
- Supported by free AVR Basic Compiler
- 30 User I/O's
- One user LED
- Single 3.3V Supply
- Easy solder: SMD or TH mountable
- Size 22.5x22.5x2 mm

Pin Configuration

Description

AP32PA3 is a low power 8-bit Microcontroller with ISA mostly compatible to Atmel Classic AVR series. The microcontroller is implemented as Soft-Processor in Actel ProASIC3 Family lower Power Flash FPGA A3P060.

Block Diagram

TODO

Pin Descriptions

GND

Ground pin. There is one extra GND pad close to the JTAG pads.

VCC

Supply voltage pin, should be connected to 3.3V supply, it is recommended to place an extra bypass capacitor close to the GND-VCC pins.

PORT (P2,P4..P32)

User port with 30 I/O Pins. All pins are configured as inputs at power on. Most pins have internal pull-up resistor enabled. Some port pins have alternate functions.

TCK

JTAG TAP TCK input, this pin is readable by the processor so it can optionally used as extra input only port.

TDI

JTAG TAP TDI input, this pin is readable by the processor so it can optionally used as extra input only port.

TDO

JTAG TAP TDO input, this pin is writeable by the processor, however the value will not be updated on the output pin unless there is external TAP master supplying TCK, and the TAP is in USER shift DR state.

TMS

JTAG TAP TMS input, never directly readable by the processor.

User I/O

There are total 30 I/O pins available. All I/O ports are configured as input at power on.

Left			Right			Top			Bottom		
Pin	**Name**	**Pull**	**Pin**	**Name**	**Pull**	**Pin**	**Name**	**Pull**	**Pin**	**Name**	**Pull**
1	**GND**	-	**17**	P17	Up	**25**	P25	None	**9**	P9	Up
2	P2	Down	**18**	P18	Up	**26**	P26	Up	**10**	P10	Up
3	**VCC**	-	**19**	P19	Up	**27**	P27	Up	**11**	P11	Up
4	P4	Up	**20**	P20	Up	**28**	P28	Up	**12**	P12	Up
5	P5	Up	**21**	P21	Up	**29**	P29	Up	**13**	P13	Up
6	P6	Up	**22**	P22	Up	**30**	P30	Up	**14**	P14	Up
7	P7	Up	**23**	P23	Up	**31**	P31	Up	**15**	P15	Up
8	P8	Up	**24**	P24	Up	**32**	P32	None	**16**	P16	Up

Table 1 Pin to I/O port mapping

Note: two additional input pins can be used from JTAG port pads.

I/O Alternate Functions

Some Ports have alternate functions that can be enabled by setting configuration registers.

Port	Function	Comment
P2	PLL GLA Output	Programmable clock
Px	PLL GLB Output	Programmable clock
Px	PLL GLC Output	? maybe use as programmable delay
P25	OSCOUT1	No pull up/down
P32	OSCOUT2	No pull up/down
P9	OSCIO1	One pin oscillator 1
P16	OSCIO2	One pin oscillator 2

Table 2 I/O Alternate Functions

The configuration hardware is limited and some resources are shared, so not all options are available at same time.

I/O Multiplexer

Each I/O pin (32 total) can be accessed by direct I/O Commands. Additionally all 32 pin inputs are connected to another "static" multiplexer. The output of that multiplexer can be connected to several function blocks. The multiplexer output can optionally be inverted.

Resource	Input	Usage
Status Register	Bit 7	As flag in short branches
PLL	CLK	Alternate input clock for PLL
Timer/Counter	Count	Frequency measurement or pulse counting
Timer/Counter	Gate	Pulse width measurement
Port I/O	P25 (out)	Two pin oscillator
Port I/O	P32 (out)	Two pin oscillator

Table 3 I/O Multiplexer connectivity

There is only one such static multiplexer so changing the pin index connected to it, will connect that pin to all functions that have selected the multiplexer output as its input. This is normally not desired, so only one function can be used at any one time.

Some functions however can be used at same time too, as example when the multiplexer is used to make a two pin oscillator then the output can also be connected to the timer/counter block for frequency measurement.

JTAG Pads/Header

JTAG pins are routed to row of TH pads, the main purpose of them is factory programming. There is no need to connect anything to them.

However two JTAG port pads (TCK and TDI) are readable by the processor and are directly mapped to the IO address space.

It is also possible to update the Flash BIOS over the JTAG port by using some JTAG cable and software capable to play back SVF files.

I/O Address Map

TODO

Address	Bit(s)	Name	R/W	
3F	6..0	SREG	R/W	Processor Status Register (Flags)
3F	7	PFLAG	R	Port Flag (output of static port input mux)
2B	4..0	PFSEL	W	Port Flag selector
2A		CR3	W	Control Register
29		CR2	W	Control Register
28		CR1	W	Control Register
20	7..0	NCO	W	Clock setting: 0x01 lowest, 0x80 highest (F/2)
20	7..0	SR1	R	Status Register
1F	0	LED	W	
1F	0	TCK	R	
1F		TDO	W	
1E	0	TDI	R	
00..1D	0	Px	R/W	Single bit I/O Ports P2..P32

Table 4 I/O Address Map

Interrupts

Processor master interrupt enable bit is not implemented (core interrupts always enabled). So enabling peripheral interrupts will start the interrupt system also.

Currently only timer tick interrupt is available at vector 0x0001.

Clock Manager

TODO

NCO

A NCO is used to generated clock for the System. The MSB of the 9-Bit phase accumulator is used as NCO clock output. Phase increment is programmable (8 bit wide), it is mapped to I/O register space. A value of 0x80 yields to max clock output setting, what is input clock divided by 2, at input clock 50MHz the maximum output clock would be 25 MHz. NCO increment register is preloaded with value 0x01 at power up, giving lowest system. Writing 0 to NCO register will yield to minimal clock (not stopped clock).

NCO	MHz	NCO	MHz	NCO	MHz	NCO	MHz
1	**0.1953125**	33	6.4453125	65		97	
2	**0.390625**	34	6.640625	66		98	
3	0.5859375	35	6.8359375	67		99	
4	**0.781250**	36	7.03125	68		100	
5	0.9765625	37	7.2265625	69		101	
6	1.171875	38	7.421875	70		102	
7	1.3671875	39	7.6171875	71		103	
8	**1.5625**	40	7.8125	72		104	
9	1.7578125	41	8.0078125	73		105	
10	1.953125	42	8.203125	74		106	
11	2.1484375	43	8.3984375	75		107	
12	2.34375	44	8.59375	76		108	
13	2.5390625	45		77		109	
14	2.734375	46		78		110	
15	2.9296875	47		79		111	
16	**3.125**	48		80		112	
17	3.3203125	49		81		113	
18	3.515625	50		82		114	
19	3.7109375	51		83		115	
20	3.90625	52		84		116	
21	4.1015625	53		85		117	
22	4.296875	54		86		118	
23	4.4921875	55		87		119	
24	4.6875	56		88		120	
25	4.8828125	57		89		121	
26	5.078125	58		90		122	
27	5.2734375	59		91		123	
28	5.46875	60		92		124	
29	5.6640625	61		93		125	
30	5.859375	62		94		126	24.609375
31	6.0546875	63		95		127	24.8046875
32	**6.25**	64	**12.5**	96		128	**25**

Table 5 NCO Frequencies

Frequencies with no jitter are shown in bold.

PLL

The PLL Clock input is connected to 50 MHz clock from the on board oscillator at power up. It can be connected to either NCO output or any I/O pin by changing configuration register values.

The PLL input frequency range is 1.5 to 350 MHz so some lower NCO frequencies cannot be used as input clock to the PLL.

PLL Output	Power up setting	Comment
Channel A	100 MHz	P2
Channel B	? MHz	P?
Channel C	? MHz	P?

Table 6 PLL Default output frequencies

VCO Gear (VCOSEL[2:1]) is set to xx at power on.

VCOSEL[2:1]	VCO min (MHz)	VCO max (MHz)
00	24	43.75
01	33.75	87.5
10	67.5	175
11	135	350

Table 7 VCO Gear frequency ranges

The PLL dynamic reconfiguration interface is connected to registers in the I/O space so it is possible to change the default values of the PLL parameters.

For complete PLL description consult Actel ProASIC3 Handbook.

AVR Compatibility

The processor core is mostly compatible to Atmel “Classic AVR” RISC Microcontroller.

Instructions/Functions not implemented

- WDR and Watchdog
- LPM – use LD to access ROM space
- AVR standard peripherals

Differences in core behavior

- CPU Interrupts are always enabled (not controlled by SREG bit 7)
- Hardware stack is 4 level deep
- Instruction ROM is directly mapped to RAM space
- I/O and Registers are not accessible using LD/ST Instructions
- CBI/SBI can write Carry or T bit with optional inversion to Port pins
- SREG bit 7 is can be used as fast I/O flag input

Performance

AP32PA3 processor core and peripherals have been optimized for best bit-bang software controlled I/O performance, both for code compactness and execution speed.

Any port pin can be set to 0 or 1 by using single instruction (CBI/SBI).

Any port pin can be set to the value of CARRY or T flag with optional inversion. Those bit copy from any register bit to any I/O port is only two instructions.

Status register bit 7 can be connected to any I/O port pin input, those making I/O pin value available for short branches.

Special Features

The processor has very little dedicated peripherals. There are however special dedicated circuits added that allow different features to be implemented with the assistance of the software.

Pulse width Measurement

Pulse width can be measured on any pin, minimum measurement step is 10ns.

Frequency Measurement

Frequency of an signal can be measured on any pin up to 200MHz.

Capacitance Measurement

Two pins (P9, P16) can be configured as one pin oscillators and the frequency can be measured, the frequency will depend on the extra capacitance added to those pins.

Inductance Measurement

There are two pins (P25, P32) that can be configured as an oscillator (internal inverted feedback), the frequency of the oscillation can be measured. The oscillator oscillates when external inductor is connected to those pins. Those two pins have no pullup or pulldown resistors enabled.

Ambient Light Measurement

There are two pins (P25, P32) without resistor pull up/down's, those pins can be used to connect a normal visible LED to allow ambient (visible) light sensing applications.

Bootstrap

A small bootstrap code is programmed to the 128 byte Flash ROM, this code will boot the actual application program from some media or interface connected to the I/O Pins. The Flash ROM contents is copied to the MCU Instruction memory at address 0 at cold boot before the system internal reset is released. Because the Flash ROM size is only 64 instructions only one type of Bootstrap can be programmed to it. The bootstrap ROM is not reprogrammable from the application code, it can only be changed by JTAG Programmer.

SD Card Boot

This loader will load one sector from SD or micro-SD card to memory and execute the loaded code. SDHC and MMC Cards are not supported. The Card is initialized in SPI mode, so it must also be used in SPI mode by the application.

UART Boot

Simple UART loader can load the instruction memory and execute the loaded code. The code can be loaded to the RAM and executed by simply sending a special ASCII sequence to the UART, no handshaking is used or needed.

SPI Master Boot

This bootstrap will load code from Microwire EEPROM or SPI Flash serial ROM.

Customizing Bootstrap

On request customized bootstrap's can be made, or optionally it is possible to reprogram the bootstrap code using a JTAG cable and tools. The best way to develop a custom bootstrap is to use the FPGA Evaluation package and some supported FPGA evaluation board. Updating the bootstrap with DATA2MEM is faster process than reprogramming the Flash ROM on the board.

Programming

AP32PA3 Core is mostly compatible to AVR Classic ISA, so any tool able to emit valid code for AVR could potentially be used. The small code space however almost fully excludes all high level language tools except the AVR Basic Compiler.

AVR Basic

This is the recommended compiler, libraries and examples are provided with the AP32PA3 SDK.

```
// P2 = pin 2, P4 = pin 4, P32 = pin 32
P2 := 1; // set port pin to  1
P2 := 0; // clear port pin to 0
```

Writing to ports pins.

```
PFSEL := SEL32; // select pin 32 as flag input
IF PORTFLAG then // use port I/O in branch instruction !
        LED := 1; // Lit on-board LED
ELSE
        LED := 0; // LED off here
END IF;
```

Using the I/O Port multiplexer to sense port value in short branch instruction. In the example above the branch on port input value uses one instruction, normally it would have used two instructions.

Demo Designs

List of available demo designs. Some designs are also offered as ready to run bitstream's for selected FPGA evaluation boards (not the AP32PA3 module).

Character LCD "Hello"

Demo is ready to run on Xilinx Spartan-3A Starterkit.

Application Examples

TODO

FPGA Configurator

The special bypass hardware makes it possible to copy bit streams from either SPI serial flash or SD card at bit rates up to 12.5 MBit/s under full software control. Of course after FPGA configuration additional features can be implemented by the program, like providing clock to the FPGA from the on board PLL, implementing a gateway to the SD Card, etc.

FPGA configuration in slave serial, slave SPI or JTAG modes can be implemented.

For this application it is normally sufficient to use the standard SD Card bootstrap, but of course a custom boot strap code could be developed as well to the specific requirements.

UART to XYZ adapter

UART or USB UART could be used to make a adapter to adapter to almost anything. UART interface consumes two I/O pins, so there would be 28 user I/O left for the program control. This type of application could bootstrap either from UART or from SD Card (depend on the bootstrap used).

Programmable PLL

TODO

Frequency Meter

TODO

Development Tools

TODO

AVR Basic Compiler

This is a special version of the original AVR Basic Compiler developed in 1997 by Silicon Studio Inc. Output file conversion utilities are provided, to generate files suitable for Actel FlashPro3 (UFC files) and Xilinx DATA2MEM (MEM files) for evaluation on Xilinx Boards.

AVRA AVR Assembler

This is a free open-source assembler compatible to AVRASM Assembler. Can be used to develop applications, include files are provided.

NoFAT™ Utility

AP32PA3 SD Card bootstrap can load code from file written to the SD Card by normal file copy, however the file need to be prepared by special tools. The NoFAT tool prepares and updates this special file.

UART Companion

Special tool to download firmware using the UART Bootstrap Flash BIOS, also works as terminal program later.

Clock Configurator

This tool allows to calculate all valid frequencies that can be obtained from the NCO and/or PLL. Settings are generated to configure the NCO and/or PLL.

FPGA Evaluation

AP32PA3 can be evaluated on general purpose FPGA Boards.

List of supported FPGA Boards

- **Xilinx Spartan-3A Starterkit**
- Xilinx Spartan-3AN Starterkit (not tested, only bitstream's are generated)
- **Trenz TE300 with Demo base Board**

The evaluation is delivered as ready to download bitstream for the given FPGA development platform, AP32PA3 port pins are hardwired to the available peripherals and/or user headers on the target board.

The actual features that can be directly evaluated depends the target board used.

S3A-SK Demo

This FPGA Evaluation package allows to test many features of the AP32PA3 using Xilinx Spartan-3A Starter kit as system emulation board. Almost all features are identical to the AP32PA3 board level product, with the exception of the JTAG and PLL features. Dynamic reconfiguration of the PLL is supported but it only changes the post VCO divider, not the settings of the PLL input or feedback dividers.

The Flash BIOS image can be updated in the bitstream by using Xilinx DATA2MEM tool (Xilinx free WebPack installation is required for this). Several readymade bitstream's are provided too with standard bootstrap code already embedded.

For test purposes the instruction RAM can also updated with DATA2MEM, the lower 128 bytes are however overwritten by the content of the Flash BIOS.

Peripherals that can be evaluated

- "On-board" LED – FPGA Init (Red LED LD11)
- UART – DB9 Board Connector J36
- SD Card (Trioflex 6-pin micro-SD adapter is required) – PMod header J18
- 4 User I/O's – PMod header J20
- 4 User I/O's – PMod header J19 (connected to the OSCxx capable pins)
- Switch inputs connected to I/O
- LED's connected to I/O
- Character LCD
- *SPI – can be connected to various on-board components*
- *Audio output – Audio jack J29 (depend if we can have reasonable DAC hw)*
- *VGA Video – HD-DB15 Board Connector*

Function	Connection	AP32	Notes
FPGA Init	Red INIT LED LD11	LED	
UART RX	J36	P2	
UART TX	J36	P4	
LCD D4		P26	
LCD D5		P27	
LCD D6		P28	
LCD D7		P29	
LCD E		P30	
LCD RS		P31	
Rotary Push		TCK	Rotary knob switch (push for “1”)
Rotary A			
Rotary B			
SPI CLK	J18	P10	For micro-SD adapter (or user I/O)
SPI MOSI	J18	P11	
SPI MISO	J18	P12	
SPI SEL	J18	P13	
	Yellow AWAKE LED	P15	
	J20	P5	User I/O
	J20	P6	
	J20	P7	
	J20	P8	
	J19.1	P9	One pin oscillator 1
	J19.2	P16	One pin oscillator 2
	J19.3	P25	Two pin oscillator
	J19.4	P32	Two pin oscillator
		P14	
		P17	
		P18	
		P19	
		P20	
		P21	
		P22	
		P23	
		P24	
		TDI	
		TDO	
Reset	SW0	-	This not available on AP32 at all.

Table 8 Pin to S3A-SK board function mapping

The mapping on S3A-SK does not fully match the default mapping for factory bootstrap or CRUVI evaluation board mappings. So the firmware tested on S3A-SK must be recompiled for actual use on real AP32PA3 boards.

TE300 Demo

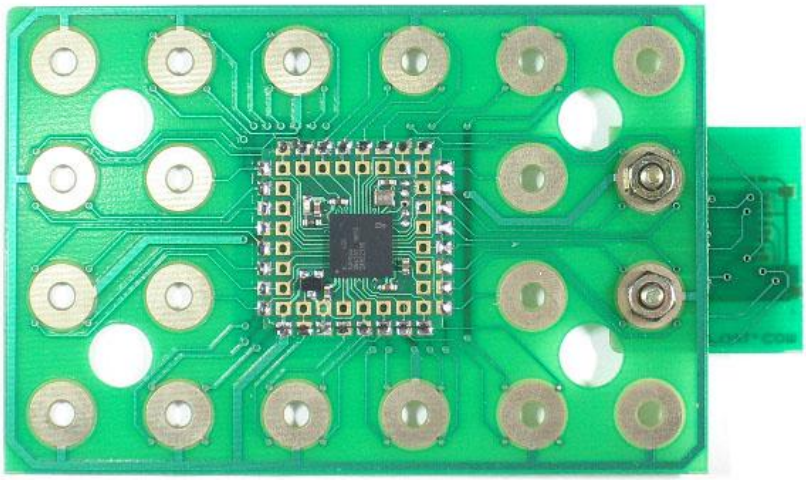
TE300 micro-module and TE300 Demo base board are used as evaluation platform.

Function	Connection	AP32	Notes
LED	TE300 LED	LED	
Push Button	TE300 Button		
UART RX			DB9 on baseboard
UART TX			
SD			Micro-SD Socket on baseboard
SD			
SD			
SD			
SPI			Serial Flash on TE300 module
SPI			
SPI			
SPI			
I^2C			
I^2C			
		P9	One pin oscillator 1
		P16	One pin oscillator 2
		P25	Two pin oscillator
		P32	Two pin oscillator

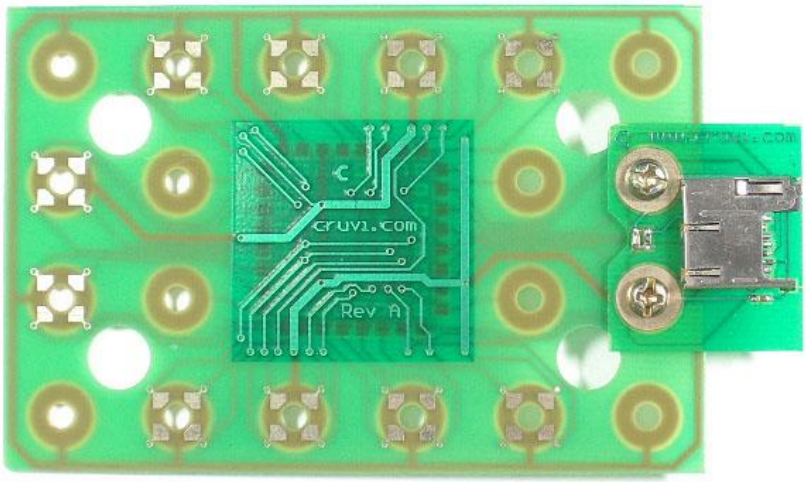
Table 9 Pin to TE300 function mapping

CRUVI™ EVAL20

Baseboard for evaluation STAMP32 form factor modules with the CRUVI™ concept modular design.



AP32PA3 soldered to CRUVI™ baseboard (bottom view)



Baseboard ready for boot from micro-SD Card (power adapter board not shown).

Licensing

AP32PA3 is normally delivered as factory pre-programmed FPGA on STAMP32 style module. The purchase price includes license for the IP Core use and tools. Some tools may have separate license and/or be limited or locked to specific usage.

IP Core license

The IP Core for the complete SoC system can also offered as VHDL source code. It is highly optimized and tuned for implementation in ProASIC3 A3P060, it can however also be used in other Actel FPGA device, or even in non Actel device. Use without modification in non-Actel FPGA would be not very reasonable as most other FPGA's have more than 128 byte initialized memory so the Flash BIOS block makes no sense.

List of Tables

Appendix A: References

TODO

- Actel ProAsic3 Handbook
- ST32PA3 Datasheet