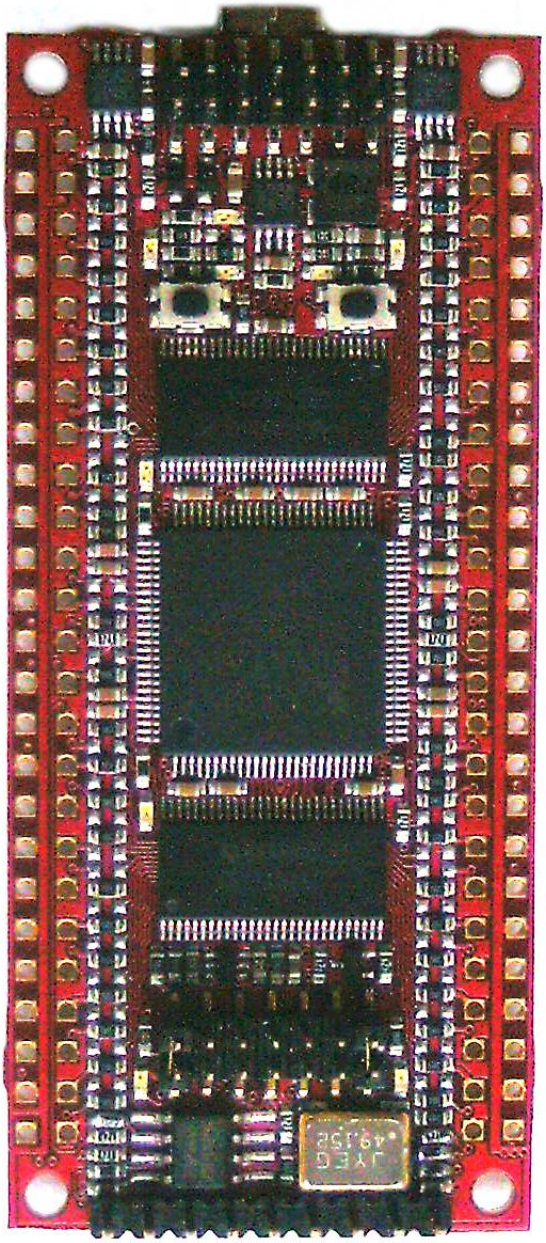
Antti-Brain

Issue 6

February 2009



Revised on February 28, 2009

Editorial

Getting ready for Embedded World 2009, it did come much faster than expected!

Antti Lukats
Antti.Lukats@googlemail.com

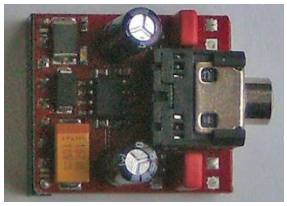
http://groups.google.com/group/antti-brain

Cover Story

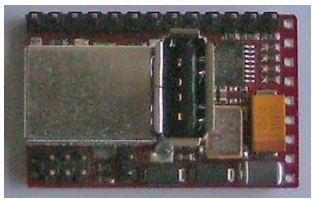
OHO GODIL48 modules are ready (soon to be available from the online shop as well). The cover page picture shows the main module, it can be fitted with XC3S100E, 250E or 500E, optional features include USB microcontroller and 5V tolerance switches.

GODIL48 modules can be mounted as DIL48, or then using 100mil dual row headers.

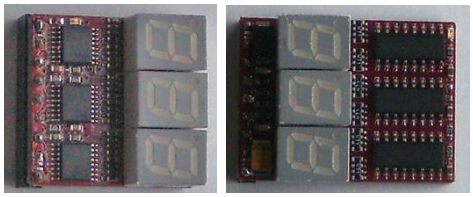
Here are some accessory boards for development from OHO, those are also useable for the GOP200 DIL module as they have the same header.



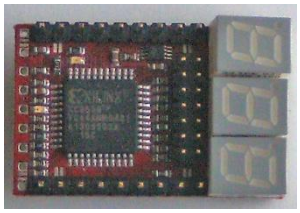
Audio Module for PWM/delta-sigma DAC type.



USB PHY Module, also includes 48MHz oscillator.



7 Segment LED modules.



7 Segment LED module with CPLD, this module can also be used as standalone board (it has option for local oscillator).

Actel Part I

For some time ago I was very disappointed about Actel tools. They are still a little bit annoying, but some features are pretty nice already.

CoreABC

ABC – **A**PB **B**us **C**ontroller. We could compare it to Lattice Mico8 or Xilinx PicoBlaze, or to Altera Avalon sequencer.

I happened to evaluate CoreABC for a project where, well I was too lazy to write a complex state machine in HDL, so I did think to give CoreABC a try. Actel FPGA's do not have RAM initialization so the CoreABC is somewhat special that its logic-ROM generator uses don't-cares inside the instructions. This saves some logic resources.

What is really weird and somewhat annoying is that there is no way to compile the code from command line. The only way is to enter the instructions in the CoreConsole editor window.

Optimization tricks

Well first need to think that instruction ROM is done from logic. So need carefully select what instructions to use. I had used RAM variable to pass parameters for a function, changing the first parameter to be in ACC, I got LC count down from 704 to 668. Because the RAMWRT needs have address as part of the instruction, while load ACC has it not, so it yields to less resources in the generated logic-ROM.

Bad idea: I did think if I use RAM variable addresses with least number of 1's it should make the ROM smaller, but it did the opposite.

Initial register mapping:
DEF ARG0 0x11
DEF ARG1 0x12
DEF ARG2 0x13
DEF ARG3 0x14
DEF CRC7 0x17
DEF TMP 0x18
Total/ROM: 668/343

Optimized one hot ram addresses, what I assumed to make smallest ROM:
DEF ARG0 0x00
DEF ARG1 0x01
DEF ARG2 0x02
DEF ARG3 0x04
DEF CRC7 0x08
DEF TMP 0x10
Total/ROM: **772**/317

Optimize attempt 2:
DEF ARG0 0x00
DEF ARG1 0x01
DEF ARG2 0x02
DEF ARG3 0x03
DEF CRC7 0x04
DEF TMP 0x05
Total/ROM: 668/303

Optimize attempt 3 (based on variable use analyzes and mind tweaking):
DEF ARG0 0x01
DEF ARG1 0x03
DEF ARG2 0x02
DEF ARG3 0x05
DEF CRC7 0x04
DEF TMP 0x06
Total/ROM: **648**/315

ADD vs. INC: Both are optional instructions. I assumed the that using only INC is more efficient. But no, it's better to use ADD 0x01 than INC, at least this is what synthesis reports did show.

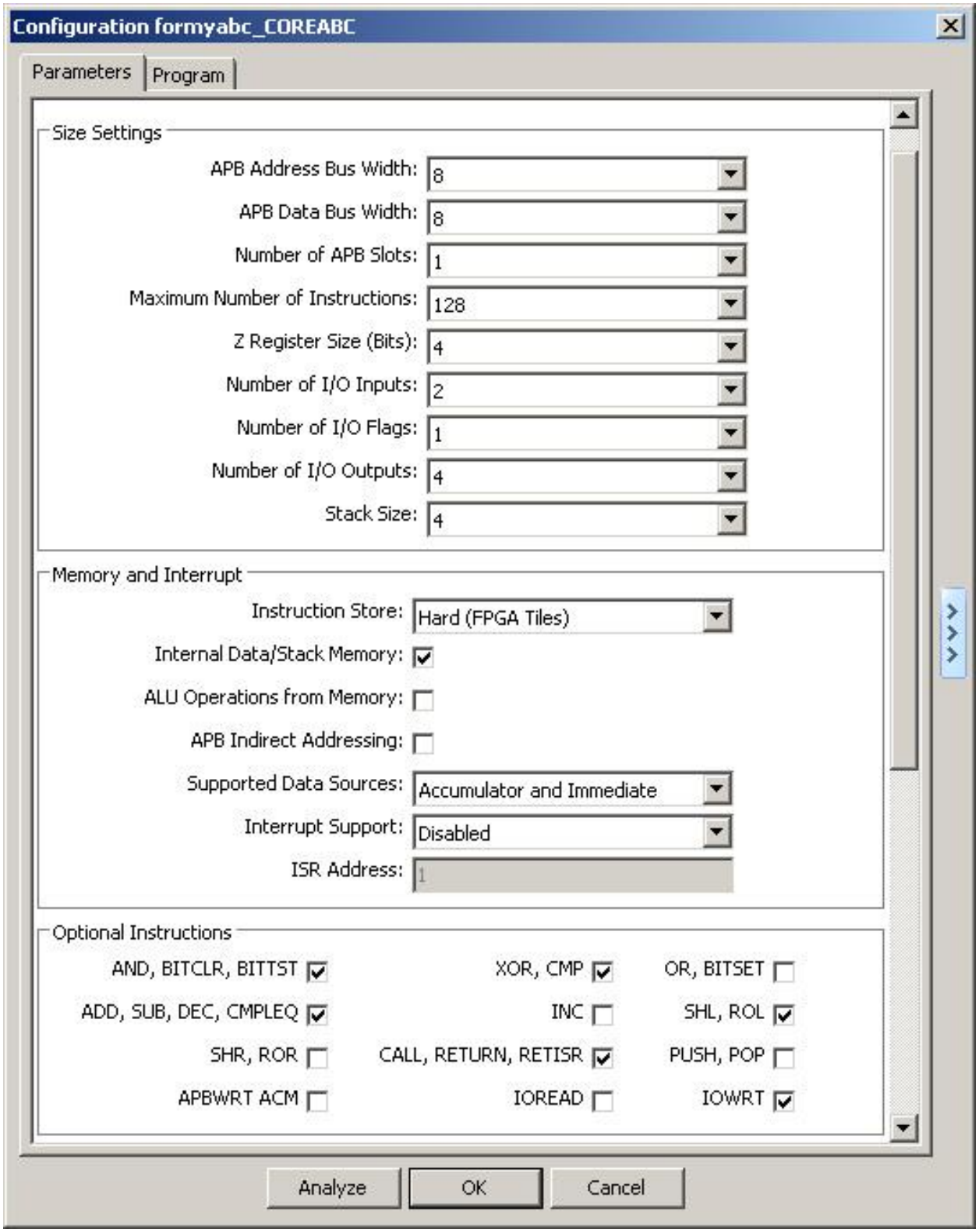
RETURN IFZERO vs. JUMP IFZERO: I assumed RETURN on condition is better, but synthesis did say the opposite!

Caution: the ROM compiler may not always give reasonable synthesis results, so some optimizations can make resource use larger without visible reason for it.

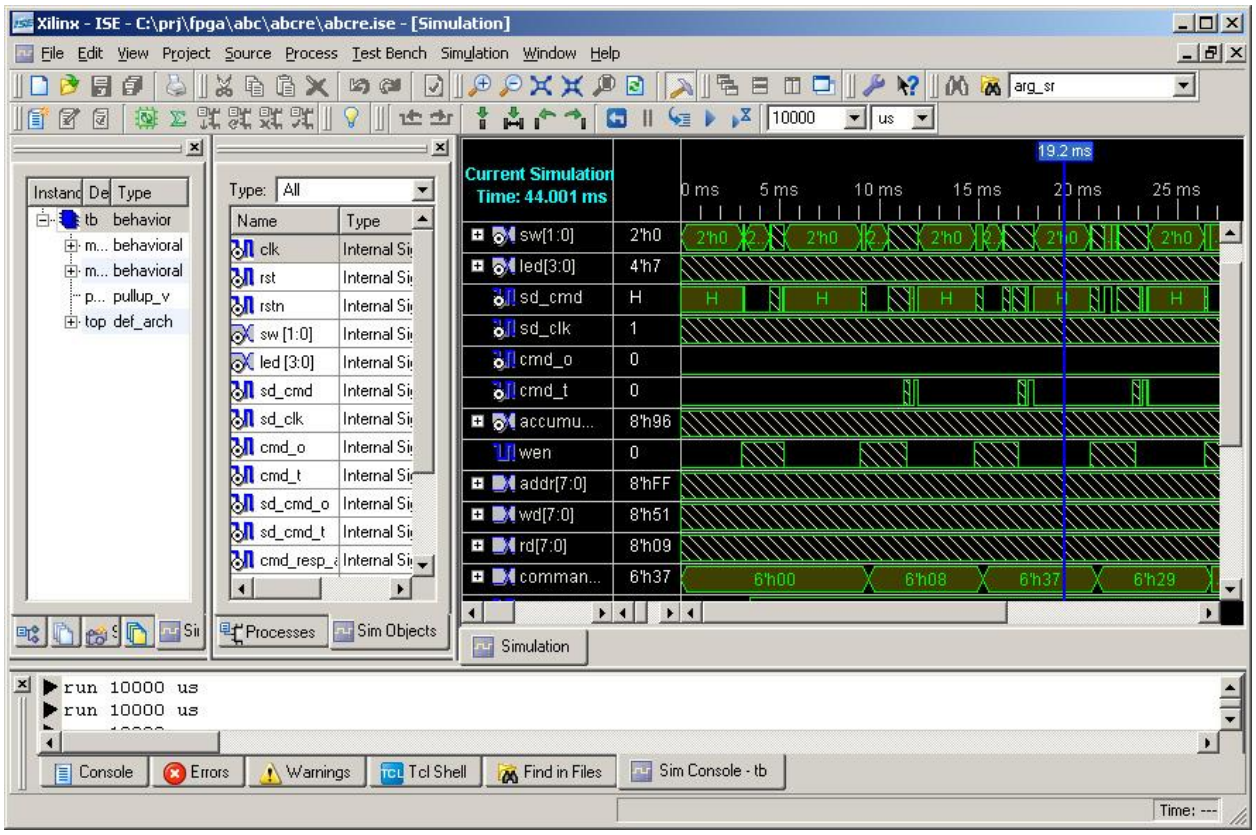
So if you want to tweak the code to yield the absolute minimal utilization, it's a lot of trial and test. Optimizing your program to have 2 instructions less, may make the logic utilization LC count larger.

Ok, no reason hiding, the state machine I am tweaking is SD Card initialization for SDHC 4-Bit mode. Current LC count is 622. It is not fully optimized or tested yet, but the simulation shows most of the init to pass as desired using my partial SD-Card simulation model.

CoreABC Configuration:



CoreABC is delivered as obfuscated RTL code. This makes it a little harder to use, but is not so big problem either. To speed up the development I created special versions that can be used with Xilinx tools, and used Xilinx ISE and ISIM for simulation.



CoreABC doing SD Card Initialization. Cursor is placed at CMD55 (0x37).

Does it really work? For initial testing I created FPGA top level design to test the CoreABC SD Card init on Xilinx Spartan-3A Starter kit. And the first bit file I downloaded did instantly work!

Before the hardware testing I had optimized SD Init program to use least number of different command constants to use less logic for Actel. Resource utilization for Spartan3A was 60 FF, 398 LUT (206 Slices). Same design for Actel, about 723 versatiles.

Controller Corner

FCU FPGA Control Unit

This just my made up name for small Soft-core processors designed for the FPGA.

	PicoBlaze	Mico8	CoreABC	Avalon Sequencer
FPGA Vendor	Xilinx	Lattice	Actel	Altera
Free License*	Yes	Yes	Yes	Yes
Deliverables	VHDL	VHDL/Verilog	Obfuscated VHDL	VHDL
Integrated to IDE	No	No	Yes	Yes*
Instruction width	18(16)	18	Variable	36
Max instructions	1024	4096	4096	64K
External Bus	Custom	Custom	APB	Avalon
I/O Address space	8 Bit	8bit + 16bit Page	Variable	32bit
Interrupt	Yes	Yes	Optional	No
Stack	Yes	16, variable	Optional, variable	Yes*
Registers	16	32(16)	ACC, Z	0 or ACC
Scratch RAM	Yes	256 Optional	Optional	No
Flags	CARRY,ZERO	CARRY,ZERO	ZERO,NEG, INPUTx	EXT, ?
Data path width	8	8	Variable	32
Cycles per Instruction	2	3	4	?
Optional instructions	No	No	Yes	No
Ready input	No	Yes	Yes	No
Instruction simulator	No*	Yes/C Source	No	No?
Assembler	Exe only	EXE/C source	Integrated	
JTAG Loader	Yes	No?	No	No?

As it comes out from the table PicoBlaze is actually least best supported! So there is still a place for a cross vendor SMALL Soft Core SoC design.

CoreABC has best IDE integration and bus-peripheral support, Mico8 has open source simulator and assembler, but for all FPGA vendors their original small soft-cores lack some nice feature.

CoreABC has not standalone assembler, PicoBlaze has no bus interface except its own, etc.

Resource comparison is not given in the table. Why? Because it really would be almost meaningless. Avalon sequencer is almost not an processor, it is really more like sequencer. KCPSM have different versions some of them are suitable for larger CLDS as well. Trying to compare the CPLD cell usage, LUT4 and versatile counts? CoreABC is much like any small soft core processor, but it can be downgraded to almost the same level as the Avalon sequencer where it is only used to initialize some APB bus peripherals.

Free license means there is no pay needed for the license. The license may however restrict the usage (vendor lock in). For PicoBlaze there is available free open-source clone called PacoBlaze.

SiLabs MCU's Part IV

Still not the easy how-to getting started guide!

Bootloader continued

My HID Bootloader, was written for IAR only, I had used IAR evaluation version to compile and test it. For a small and minor change I did compile it after the evaluation expired. Well as the code does fit 4K KickStart limit, so I install the free KickStart edition. But for some reason it did not work when compiled with the 4K limited compiler.

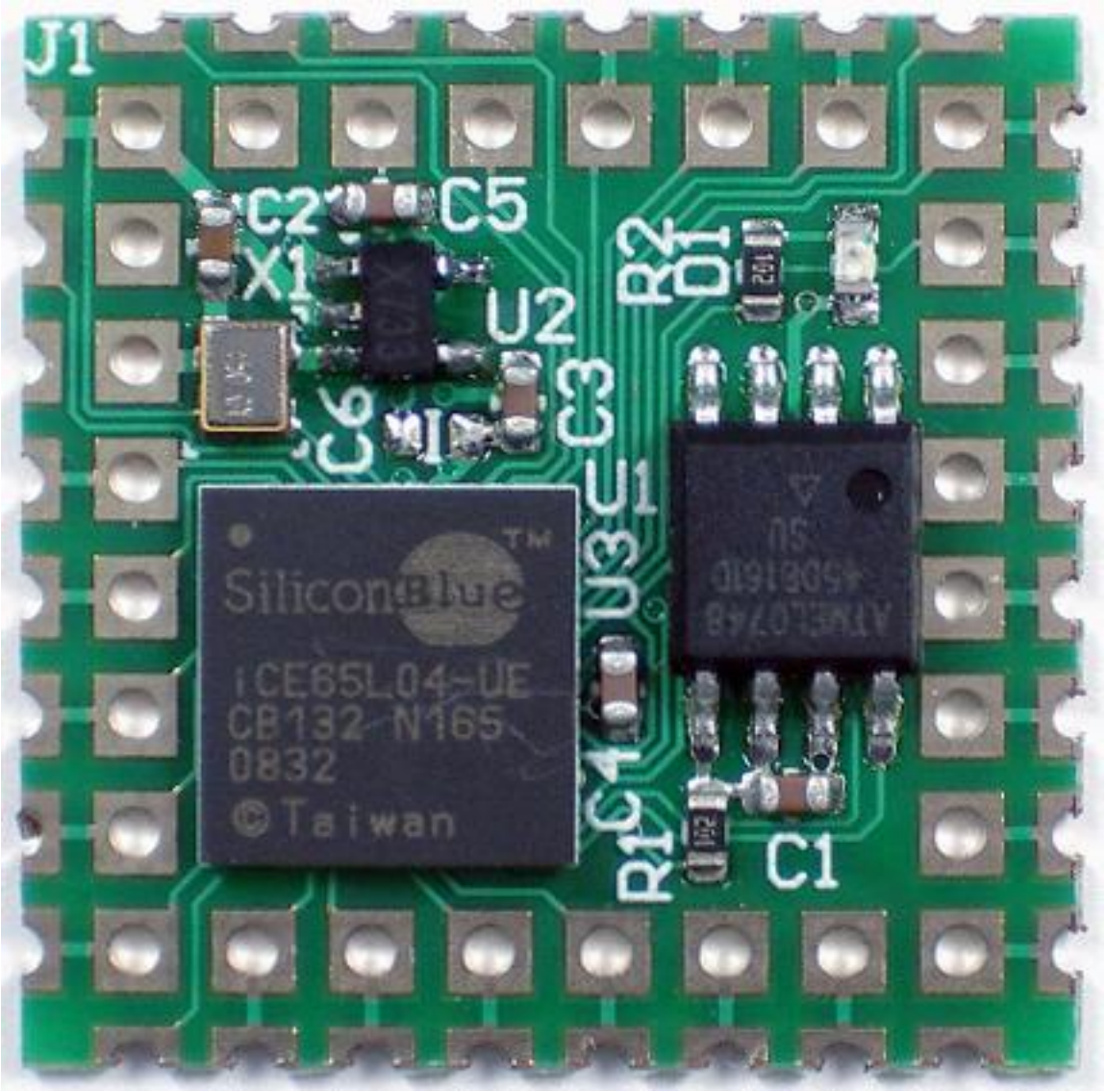
To cut the long story short: I tested the Bootloader using "led blink" application. For that I had made 2 versions led blink Keil, and led blink IAR Compiler. The C code in both directories was named "blinky.c". Both folders did include "blinky.bin", one of the folders did include also "blinker.bin". Now in all my testing I used "blinky.bin" from the folder that also had the "blinker.bin". And that file I used happened to be 0 byte long. It did take me a week to see that. ☹ very embarrassing. All other bin files would have been Ok, only not the one I repeatedly used in my testing.

So it all works now. ☺ using it for some time already and it really is nice. No drivers to install. Just download new firmware if you need it.

But I should urgently make a more production software for the PC side download application. One that warns you if you try to make a 0 bytes update!

Silicon Blue FPGA Part II

Where is part I you may ask, well look back into the first issue, and consider that part I, so this is part II.



In the next issue…

Xilinx I

<surviving Xilinx>

MicroBlaze Debug

First: it is recommended to try use BSB or existing reference design first. When generating new system, always select "On-Chip H/W Debug Module", if not needed you can remove it later. Ok, in 10.1 the MDM connection seems to be more simple, if done later on, but I still recommend to include MDM at system generation in all cases where the target hardware has JTAG port useable.

EDK Revisions ☹

Opening Genode FX EDK project (EDK 9.2) with EDK 10.1, and

```
ERROR:MDT - Unrecoverable error(s) were encountered while updating your project
```

Not funny. Well after that error, I try open the same project file again. And it opens without problem. But trying to build I get errors that IP's are not supported on Virtex-II. Well the board has Spartan-3A?

Ok, I fix the wrong device, run build, the design compiles, but VGA display is blanked. DDR2 memory seems to be dead. So the revision updated version doesn't work.

Ok, let's take the reference from Xilinx, test the memory, and proceed? Well there is no 10.1 design at Xilinx for S3A starter board. So there is no known project with MPMC for this board.

Ok, let's create new design with BSB just to test the DDR2 memory. Failed as well. Hm.. I think my board is rev C, but BSB only supports rev D, could that be the issue?

The only information about rev C vs. D I am able to find is a note that silkscreen painting is different. I bet that should not have impact on DDR2 memory.

Hum… genode-fx precompiled bitfile also reads 0x0000 from the DDR2 memory on my Starterkit board. So maybe it has some hardware issue.

Yes it looks my S3A starter kit board has hardware problem with DDR2, so it not issue with the Xilinx design. Well, yes the 1.8V supply is missing, that explains why the DDR2 read 0.

Character LCD with MicroBlaze

I wanted to test something simple. I have plenty of Xilinx boards, but none of the boards that have on board JTAG have USB UART's and my notebook has no COM port. Sure I could take some board with USB-UART like Memec S3-1500 board, but then I would need to drag the USB-UART adapter with me. Ok, I decided to use Xilinx Spartan-3A Starterkit. This board is currently actively marketed by Xilinx, so I hoped for best support for this board.

As I only need small debug print, I decided to use the onboard character LCD. I have done char LCD stuff for different Xilinx boards myself using different IP or just GPIO bit-bang. But as the Xilinx reference design uses LCD, I decided to use Xilinx solution as I believed it is the fastest way to success. I look for reference designs, with MicroBlaze and LCD. Ok, the only one available is for web server demo for EDK 9.2, so I download that. I add to my own system the LCD IP, I copy the UCF file. I copy Xilinx LCD function library and the LCD initialization and hello code into my main code.

Should work out of the box?

No way. I spend some hours checking the parameters, UCF file, connections, no luck. All seems correct but there is no greeting on the LCD.

Ok, I think let me take the unmodified web server system and try that. Without any changes it should display something on LCD?

Wrong. EDK core generator self-terminates and the system build process stops. ☹

Simple thing. Get some print to LCD. The Xilinx way however starts with frustration.

Using GPIO and my own bit-bang code would have been faster. Even rewriting the LCD bit-bang would have been faster. I just wanted to use Xilinx known working solution. That didn't work.

ISE/EDK 11.1 are soon to be launched, but Xilinx has not a single 10.1 EDK design for the S3A-SK, and the 9.2 design after rev up crashes in core generator. Nice.

Second try, looks like this time it passed core generation… so just need wait if it the implementation flow completes.

Well, it still runs, but already have DIRT failure:

```
INFO:Route - One or more directed routing (DIRT) constraints generated for a specific
device have been found. Note that
   DIRT strings are guaranteed to work only on the same device they were created for.
If the DIRT constraints fail,
   verify that the same connectivity is available in the target device for this
implementation.

# of EXACT MODE DIRECTED ROUTING found:1, SUCCESS:0, FAILED:1
```

Sure I only wanted to test LCD, so I do not need the DIRT or DDR2, but hey I wanted to test Xilinx original design, so I wanted to compile it AS IS without modifications.

Ok, next error:

```
Copying Library Files ...
ERROR:MDT - issued from TCL procedure "::sw_cpu_v1_11_b::generate" line 229
    cpu () - The software settings parameter CORE_CLOCK_FREQ_HZ is deprecated
   and its value (66670000 Hz) in the MSS is inconsistent with the processor
   frequency inferred from the hardware (66666667 Hz). Please remove the
   parameter CORE_CLOCK_FREQ_HZ from the MSS file.
ERROR:MDT - Error while running "generate" for processor microblaze_0...
make: *** [microblaze_0/lib/libxil.a] Error 2
```

Ok this is easy to fix. The frequency value was rounded differently. Fixed. Now removing the Ethernet from system tests software application (to make it fit into BRAM's). Update bitstream download, success! But again nothing on the LCD.

Trying to run XMD, but it doesn't see the MicroBlaze. Ok I forgot to check the linker script. Regenerating. Clean software some more trial. And no luck, nothing works.

It probably would work with EDK 9.2, but I have no intentions for the simple test of Xilinx own OPB_LCD IP core to install old version of EDK.

I am reverting to single-LED debug as the Xilinx LCD IP/driver stuff doesn't work without debugging.

Hm.. maybe the LCD is broken? I load Ken Chapman's DNA Reader demo bit file. This shows LCD display so the LCD is OK.

Well I try my own design again.. and suddenly the LCD display printout works!

But same design a few hours ago didn't? Ok, just a advice, if something doesn't work, keep trying! Sure it is the question how long… this time I have no explanation why it didn't work first time.

Well it is solved the puzzle. The LCD clear command needs additional delay in software before any characters can be written to the LCD. Simple as that. ☺

THE EASY WAY

MicroFPGA solution, write 2 lines of C code:

```
LCD_Init();
xil_printf("wwwMICROFPGAcom");
```

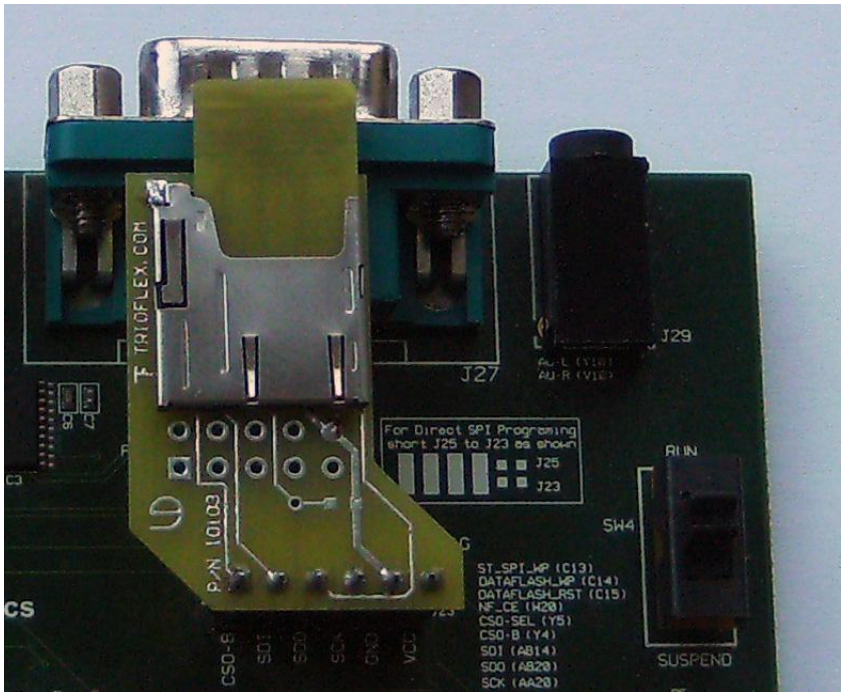
And the text is displayed on LCD of the Spartan3A starter kit ☺ Well I really need to work again on the MicroFPGA! Please note, that the MicroFPGA flow does not include using Xilinx tools. Just write C code!

Single Sided

Simple electronics die hard.

UD SPI Flash, AT45

This is the second PCB version already still not perfect. This board is made as single sided not because cost only, but because it makes the other side completely flat. Of course the PCB price is little lower as well, but the most expensive about this PCB is funnily the routing, it has to be done with 0.8mm router.



This is how the UD SPI Flash card is used to configure Xilinx Spartan-3A FPGA. If the adapter board is inserted the other header row then Impact can be used to program the SPI flash on the Card, so the S3A Starterkit works as UD SPI Flash Programmer solution as well.

References

- http://www.trioflex.com

Instead of adding the URL links at the end of each issue, I will be adding them to the Trioflex online link collection, so they can be updated more frequently.