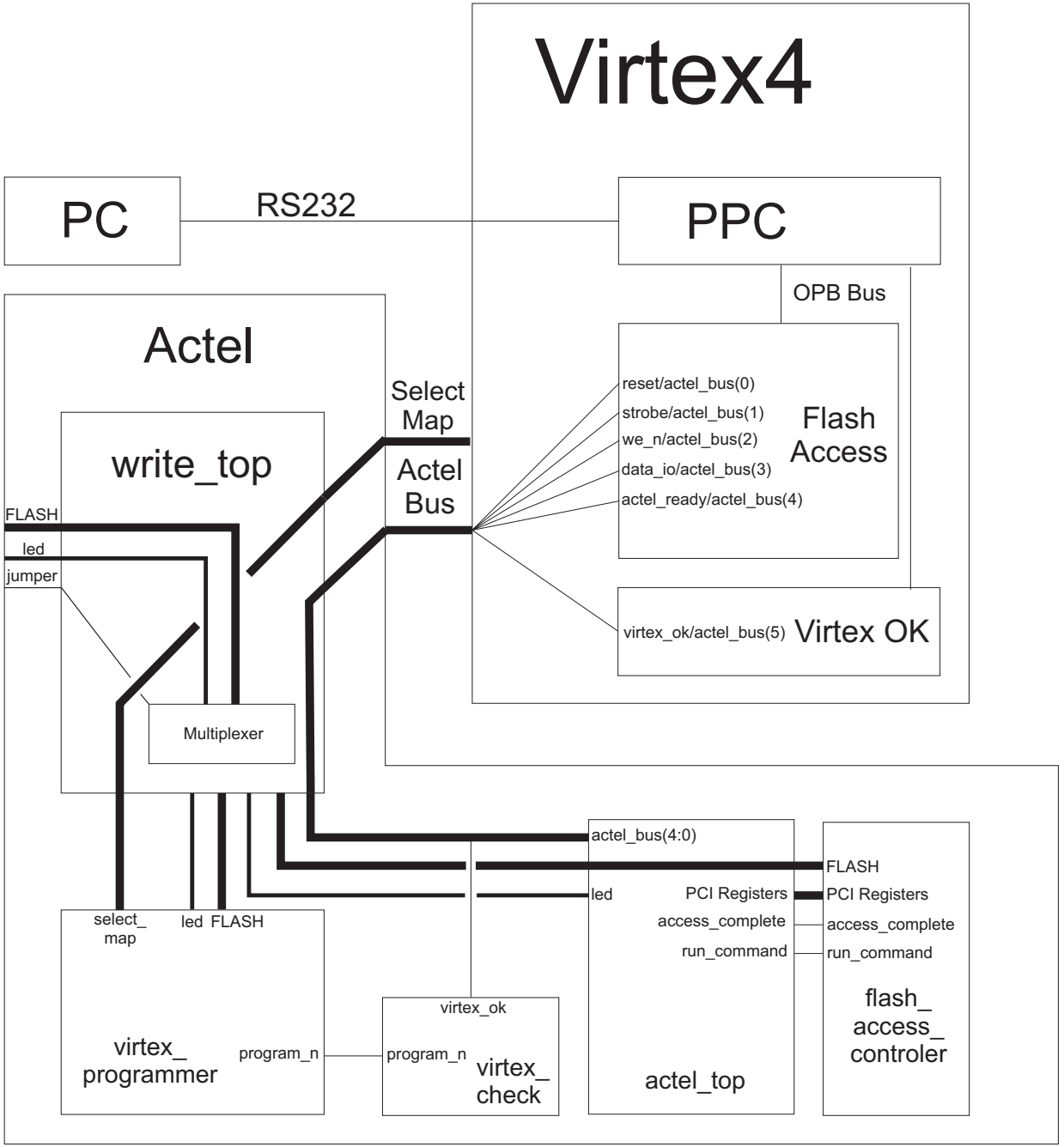
Syscore
Actel Flash Interface
Automatic Virtex Programming

David Rohr

30.04.2008


Virtex4
PC
RS232
PPC
OPB Bus
Actel
Select Map
Actel Bus
reset/actel_bus(0)
strobe/actel_bus(1)
we_n/actel_bus(2)
data_io/actel_bus(3)
actel_ready/actel_bus(4)
Flash Access
write_top
FLASH
led
jumper
virtex_ok/actel_bus(5)
Virtex OK
Multiplexer
actel_bus(4:0)
FLASH
led
PCI Registers
PCI Registers
access_complete
access_complete
run_command
run_command
select_ map
led
FLASH
virtex_ok
virtex_ programmer
program_n
program_n
virtex_ check
actel_top
flash_ access_ controler

Virtex Components

- flash_access
- virtex_ok

Virtex: flash_access

Handle communication between RS232 and Actel

- Pins
  - data_io (inout)
  - actel_ready (input)
  - we_n (output)
  - strobe (output)
  - reset (output)
- OPB Bus (myipif)

Virtex: virtex_ok

Send control clock to actel

- Pins
    - virtex_clk (output)
- OPB Bus (myipif)

Actel Components

- write_top
- actel_top
- flash_access_controller
- virtex_check
- virtex_programmer

Actel: write_top

Multiplex between Flash Access and Virtex Programming

- Pins
  - smap_data / actel_bus (pass through)
  - flash_pins / led_pins (multiplex)
  - jumper (input)
  - reset_flash / reset_programmer (output)

Actel: actel_top

Translate between Virtex and flash_access_controller PCI interface

- Pins
  - reset (input)
  - flash_pins (pass through)
  - actel_bus (input)
    (we_n, strobe, data_io, reset)
  - pci_register (output)
    (address, data, command)

Actel: flash_access_controller

Translate between PCI commands and flash interface

- Pins
  - flash_pins (inout)
  - pci_register (input)

Actel: virtex_check

Check if Virtex is running

- Pins
  - virtex_clk (input)
  - program_n (output)

Actel: virtex_programmer

Programm Virtex

- Pins
  - reset (input)
  - program_n (input)
  - smap_data (inout)
  - flash_pins (inout)

Jumper Settings

• Open / Open	Reset
• Open / Closed	Enable Flash Interface
• Close / Open	Check and program Virtex from flash chip 1
• Close / Close	Check and program Virtex from flash chip 2
Partial refresh mode:	(Parameter in Actel Design)
• Close / *	Program Virtex from chip 1 and reprogram from chip 2

Write Cycle

- Serial data transfer
- 1 bit transfered with rising edge of strobe
- 8 bit address / 8 bit data

- PCI Register Addresses:
  - 01 : Data Register (8 bit)
  - 02 : Address Register (bit 7:0)
  - 03 : Address Register (bit 15:8)
  - 04 : Address Register (bit 23:16)
  - 05 : Command Register (4 bit)

Example: Write 0xA0 to Address 0x0000FF of Flash State Machine

- Write 0xA0 to 0x01
- Write 0xFF to 0x02
- Write 0x00 to 0x03
- Write 0x00 to 0x04
- Write 0x02 to 0x05 (PCI_COMMAND_WRITE, write to command reg will start PCI command)

Write Cycle

Flash State Machine Write Sequence:

- Write 0xAA to 0xAAA
- Write 0x55 to 0x555
- Write 0xA0 to 0xAAA
- Write [DATA] to [ADDRESS]

Composition with PCI Write Cycle:

Write 0xAA to 0x01, 0xAA to 0x02, 0x0A to 0x03, 0x00 to 0x04, 0x02 to 0x05
Write 0x55 to 0x01, 0x55 to 0x02, 0x05 to 0x03, 0x00 to 0x04, 0x02 to 0x05
Write 0xA0 to 0x01, 0xAA to 0x02, 0x0A to 0x03, 0x00 to 0x04, 0x02 to 0x05
Write [DATA] to 0x01, [ADDRESS] to 0x02 – 0x04, 0x02 to 0x05

Read Cycle

- Serial read process
- 8 bit data, no address
- Direct flash access, not talking to state machine

Virtex Programming

- Trigger prog_b (chip reset)
- Wait for init_b (reset complete)
- Clock one cycle (synchronization)
- Pull cs_b down (Start programming)
- Clock out binfile byte for byte
- Release cs_b (binfile finished)
- Clock untin done high (Startup Clock)

Data transfer

Implemented in C++ Program

- Single byte (1 byte / sec)
- Burst mode (4 kbyte / sec)
- Copy from DDR RAM to Flash (10 kbyte / sec)

(Burst mode and DDR RAM copy are handled by c program in PPC)