Zylin CPU

yvind Harboe, General Manager, Zylin AS

Motivation

- Provide a 32 bit core with standard tools while taking up a minimum of real estate
- Flexible core and architecture that can be used with any FPGA brand/type, memory and IO subsystem
- Leave FPGA resources to HDL paradigm

What the ZPU is not

- Not ARM9, XScale, etc. type CPU
- Not for sustained processing
- Not for running Linux no MMU
- Unlikely to run uCLinux

ZPU - performance

- DMIPS popular performance index for deeply embedded processors
- Easy to measure with working GCC and Dhrystone v2.1
- 1 10 DMIPS depending configuration and memory subsystem
- The ZPU is for light administrative tasks
- The ZPU punches above its weight
  - Leaves resources to HDL paradigm
  - eCos performance tests are relatively better and more relevant

ZPU real estate

- LUT - FPGA MIPS. A popular index of real estate usage
- Must be measured using actual tools and part
- Xilinx XC3S400 440 LUT with 32 bit data path

ZPU - codesize

- 80% of ARM7 Thumb. Measured on eCos applications.
- There exists no MIPS equivalent for code size

ZPU programming model

- Standard programming model
- A single linear 32 bit address space
- Unused address bits are optimized away by synthesis optimisation
- IO is memory mapped

GCC toolchain

- GCC
- GDB
- newlib/libgloss   OS less applications
- Zylin Embedded CDT debug GUI

SoPC

- System On a Programmable Chip
- ZPU core running
  - eCos
  - Applications and drivers (TCP/IP, Flash-file system,...)
- Memory interfaces (Internal/external memory)
- Peripheral interface/controllers
  - Ethernet MAC (opencores)
  - Standard 16550 or simple UARTs (opencores)
  - DDR SDRAM controller (Zylin)
  - I2C, USB, SDIO, GPIO.......

eCos RTOS Real Time OS

- eCos open source free deeply embedded operating system
  - eCos is not Linux
  - Small footprint
  - TCP/IP
  - USB function drivers
  - Flash drivers
  - GDB stubs target debugging via simple UART
  - The list goes on and on
  - eCos is the rising star for deeply embedded

eCos HAL

- A ZPU comes with a variant eCos HAL configured to a ZPU port and peripherals
- Special drivers? Drivers are CPU independent.
- Opencores ethermac driver runs unmodified on SPARC, ARM and the ZPU
- Check eCos repository for your peripherals/driver stack

ZPU HDL

- ZPU's key strength is small footprint
- Starts at 300-500 LUT
- 16/32 bit datapath
- Core choice depends on memory and IO subsystem
- Dual port, single port, wishbone, custom

Custom ZPU

- Size optimisation for particular memory and IO subsystem
- More instructions? There are various instructions that can be enabled and disabled without changing compiler or binaries
- Even more instructions? May require GCC support
- Special machine code only instructions? Instructions tightly integrated with peripherals?
- Peripherals?

ISS   instruction set simulator

- Simulates ZPU to run binaries from GCC
- Takes simulator trace output for verification and cycle accurate statistics/debugging
- GDB server interface
- Simple peripheral simulation   timer, UART, etc.
- Does not replace debugging real target
- Custom ZPU verification/profiling
- Early development/training before target PCB is available
- Used for debugging eCos HAL

ZPU   roadmap

- Stay away from 50-500 DMIPS range
- Reduce footprint   obviously not a large potential
- Performance increases without sacrificing area or code density
- Improve code density   possibly at FPGA resource cost, SRAM and flash is real-estate too
- Improve GCC code quality
- Improve debug tools
- Optional peripherals?

ZPU summary

- 1-10 MIPS
- 16 or 32 datapath
- Codesize 80% of ARM7 Thumb
- GCC toolchain
- eCos HAL
- Various HDL cores available
- 400 LUTs 32 bit datapath

Questions?