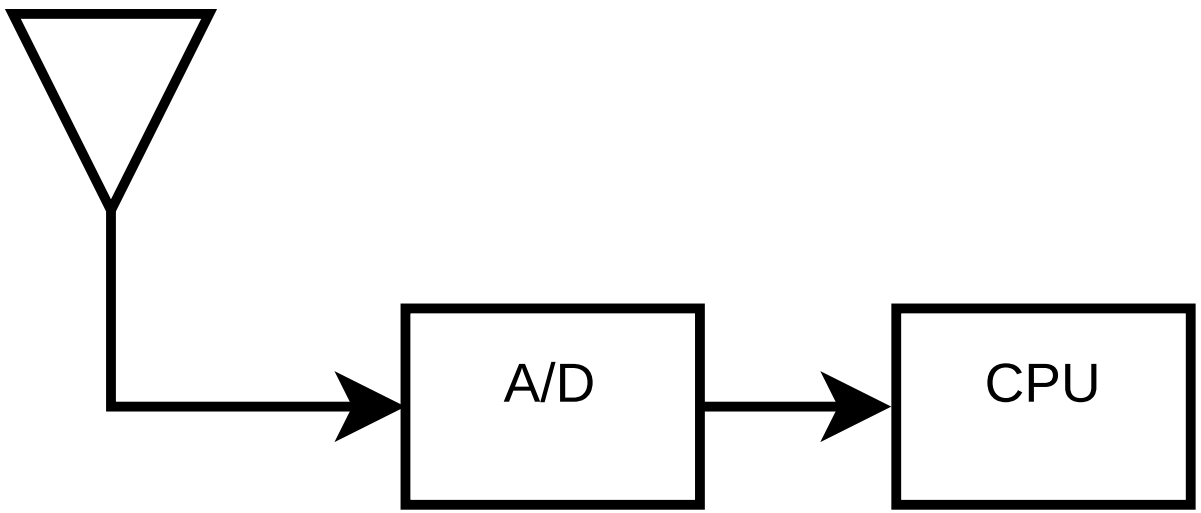
High Performance Interface between the OMAP3 and an FPGA

Philip Balister
philip@opensdr.com

Open SDR

June 23, 2011

What is Software Defined Radio (SDR)



OMAP3

- High performance (and easily available) ARM.
- 600 MHz Cortex-A8
- NEON and VFP extensions for floating point acceleration
- POWERVR^{TM}graphics hardware
- C64X+ DSP

OMAP3 Block Diagram

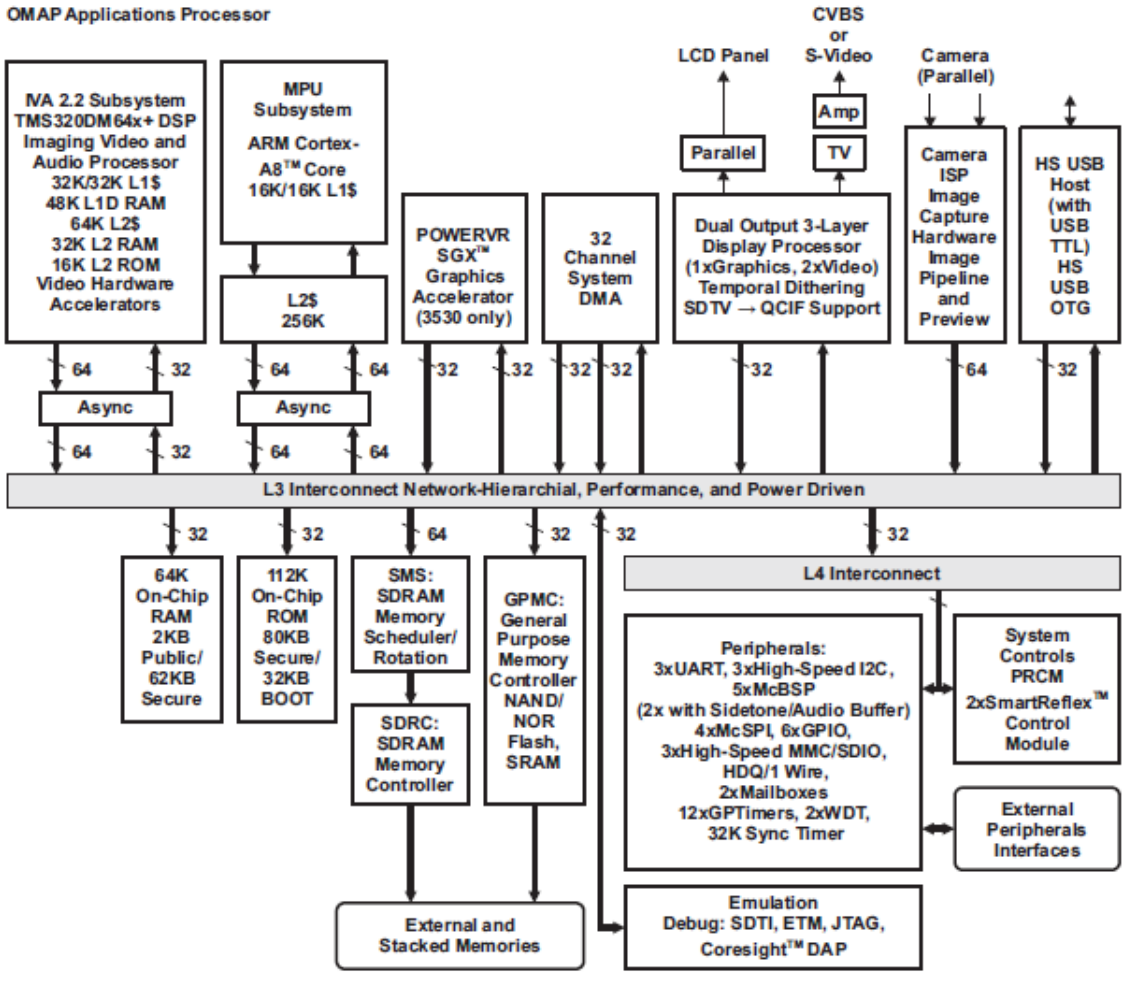


Figure 1-1. OMAP3530/25 Functional Block Diagram

ARM Cortex A8

- ARM version 7 instruction set (armv7a)
- Thumb 2 instructions allow smaller code size
- NEON SIMD coprocessor (single precision vector floating point)
- Supported by GNU Compiler Collection (GCC)
- C, C++, Java, FORTRAN, ADA etc
- Current GCC does not optimize NEON well
- Commercial compilers are available with better NEON Support

Texas Instruments C64X+ DSP

- 32 bit fixed point processor
- VLIW based instruction set
- 8 instructions per cycle, 8 execution units
- Memory Management Unit (MMU) available
- TI Code Composer for software development
- Command line Linux compiler available
- DSPLINK provides communications between DSP and ARM

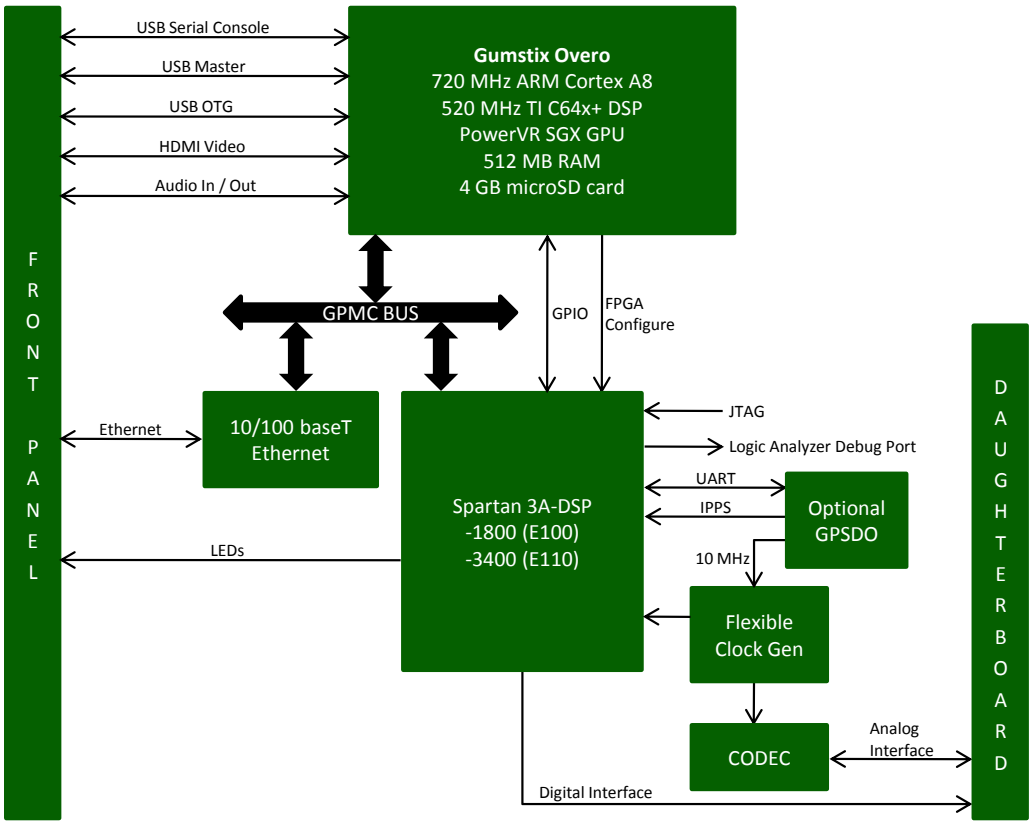
Key Features of the USRP E1XX

- High performance interface to FPGA based on GPMC controller
- Uses Gumstix Overo COM (Computer On Module) for processor
- All functions in one box, capable of standalone operation
- Develop flow graphs and block directly on the box
- Supported peripherals: monitors via HDMI/DVI, keyboard and mouse via USB
- Available with the largest Spartan 3A-DSP available.
- Same code runs on Linux, Mac, Windows, and USRP-Embedded.
- Supports all USRP daughterboards
- Open Source FPGA and drivers

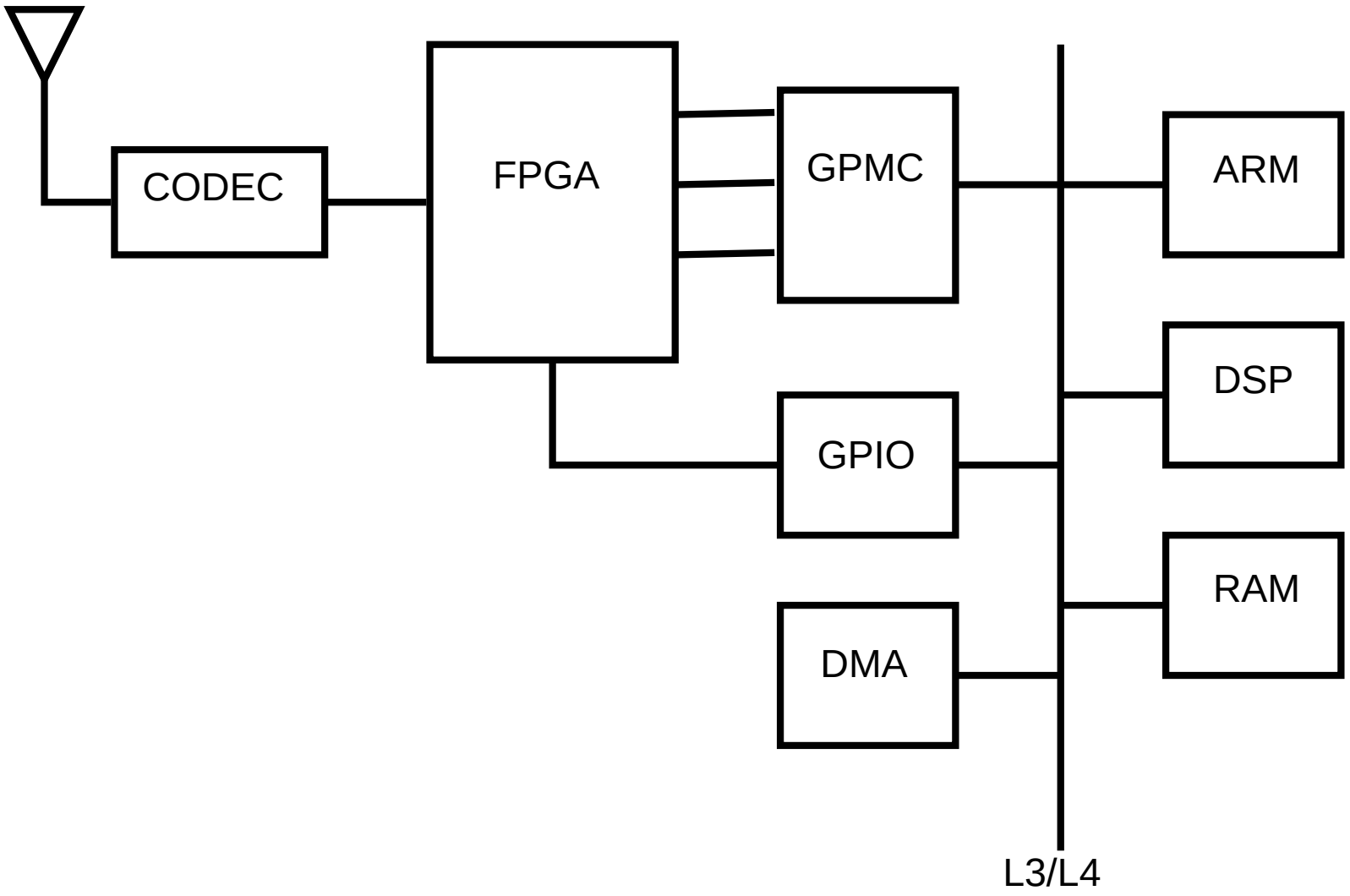
USRP E100

USRP Embedded Block Diagram

USRP E1x0 Block Diagram



System Diagram



Device Driver Overview

- Must support high data rates (MSPS)
- Zero copy design based on a large ring buffer
- DMA used to transfer data to/from the ring buffer
- User space memory maps the ring buffer
- Ring buffer allocated via get_free_pages

Ring Buffer Structure

```
struct ring_buffer_info {
        int flags;
        int len;
};

struct {
    struct ring_buffer_info tx_rbi[N];
    tx_buf[2048][N];
    struct ring_buffer_info rx_rbi[N];
    rx_buf[2048][N];
};
```

Filling the ring buffer from the FPPA

- GPIO signals the FPGA has data ready to transfer
- Read the number of bytes to transfer from the FPGA control space
- Configure the transfer destination address and length
- Start the DMA transfer
- Use dma_sync_single_for_device to invalidate the cache
- When DMA transfer ends, wake up the data received work queue
- Attempt to start a new transfer

Writing the ring buffer to the FPPA

- User writes data into the ring buffer
- User calls write to trigger data transfer from ring buffer to fpga
- Write the number of bytes to be transferred to the FPGA control space
- Configure the source address and length
- Execute data store barrier
- Start data transfer
- Wake up space available work queue
- Attempt to start a new transfer

Poll implementation

- Poll operation is critical for syncronizing user application with FPGA.
- Two wait queues monitored for wake up events (transmit and receive)
- For read, kernel has pointer to next empty entry in ring buffer ...
- For read, poll checks buffer before for valid data
- For write, kernel has pointer to next buffer to to send ...
- For write, poll checks the block before the pointer for space available

User space interface

- Flags indicate if data is available or space available in ring buffer
- If data available process data from ring buffer
- When processing complete or buffer filled, update flags
- If no data available or no space available, call poll to wait
- We can mark a buffer flag as returned from poll, but not processed

GNU Radio on the OMAP3

- GNU Radio was originally developed for high end X86 systems.
- GNU Radio does run on the OMAP3
- GNU Radio floating point performance poor
- Solution: Rewrite performance critical sections using NEON
- USRP1 works with OMAP3 system, USRP2 does not (no Gig-E solutions)

References

- https://github.com/balister/linux-omap-philip
- http://gnuradio.org
- http://www.ettus.com

Questions

Questions?