FA0 - i.MX28 + LTIB

This course describes the i.MX28 processor family and Linux Target Image Builder tool

Objectives

- This course has 4 main objectives:
  - Describing the hardware implementation and highlighting the pitfalls
  - Describing the ARM926EJ-S core architecture
  - Describing the units which are interconnected to other modules, such as clocking, interrupt controller and DMA controller, because the boot program generally has to modify the setting of these units
  - Describing independent I/O modules and their drivers.
- Note that this course has been designed from the architecture of the most complex iMX28, the i.MX287
  - Consequently, a chapter has been designed by Acsys for each possible integrated IP
  - According to the actual reference chosen by the customer, some chapters may be removed.
- Products and services offered by ACSYS:
  - ACSYS is able to assist the customer by providing consultancies. Typical expertises are done during board bringup, hardware schematics review, software debugging, performance tuning.
  - ACSYS has also an expertise in Linux porting.

This document is necessary to tailor the course to specific customer needs and to define the exact schedule.

Prerequisites and related courses

- This course provides an overview of the ARM926EJ-S core. Our course reference R1 - ARM7/9 implementation course details the operation of this core.
- The following courses could be of interest:
  - USB Full Speed High Speed and USB On-The-Go, reference IP2 - USB 2.0course
  - Ethernet and switching, reference N1 - Ethernet and switchingcourse
  - IEEE1588, reference N2 - IEEE1588 - Precise Time Protocolcourse
  - CAN bus, reference IA1 - CAN buscourse
  - Memory cards, reference IS2 - eMMC 5.0course

Course Environment

- Theoretical course
  - PDF course material (in English) supplemented by a printed version for face-to-face courses.
  - Online courses are dispensed using the Teams video-conferencing system.
  - The trainer answers trainees' questions during the training and provide technical and pedagogical assistance.
- At the start of each session the trainer will interact with the trainees to ensure the course fits their expectations and correct if needed

Target Audience

- Any embedded systems engineer or technician with the above prerequisites.

Course Outline

ARCHITECTURE OF I.MX28 MCUs

- ARM core based architecture
- Interconnect, indicating the features of AXI, AHB and APB
- Clarifying the internal data and instruction paths
- Integrated memories
- SoC mapping

THE ARM926EJ-S CORE

- CPU block diagram
- V5TE software architecture
- Memory Management Unit
- Instruction and data caches
- Debug facilities

INTERRUPT COLLECTOR

- 128-bit vectored interrupt collector to generate core IRQ
- Non-vectored interrupt collection mechanism to generate core FIQ
- Implementing interrupt nesting
- Generating the vector address

RESET, POWER AND CLOCKING

- Power Management Unit
- Reset
- Clock generation subsystem

INTERNAL INTERCONNECT

- Explaining the global AHB / AXI / APB interconnect organization
- 3-layer AHB parameterizing
- PL301 AXI crossbar parameterizing
- Dual APB
- DMA

HARDWARE IMPLEMENTATION

- Power pins
- Pinout
- GPIO module

INTEGRATED MEMORIES

- Internal SRAM
- Integrated Mask-Programmable On-Chip ROM
- On-Chip One-Time-Programmable (OCOTP) ROM

MEMORY INTERFACE

- External Memory Interface
- DDR-2 controller

- General Purpose Media Interface

TIMERS

- Timers and rotary decoder
- Pulse Width Modulator channels
- Real-Time Clock

ANALOG MODULES

- Low-Resolution ADC (LRADC) and Touch-Screen interface
- Single Channel High Speed ADC (HSADC)

SECURITY AND INTEGRITY

- Security Features
- Data Co-Processor (DCP)
- Customer-Programmable One-Time-Programmable (OTP) ROM
- 20-Bit Correcting ECC Accelerator (BCH)

CONNECTIVITY AND COMMUNICATION

- Synchronous Serial Ports
- UART
- I2C
- USB
- Fast Ethernet with IEEE1588
- 3-port L2 switch
- Dual Serial Audio Interface (SAIF), Three Stereo Pairs
- SPDIF transmitter

USER INTERFACES

- Highly Flexible Display Controller (LCDIF)
- Pixel Processing Pipeline (PXP)

GENERATING THE LINUX KERNEL IMAGE

- Introducing the tools required to generate the kernel image
- What is required on the host before installing LTIB
- Common package selection screen
- Common target system configuration screen
- Building a complete BSP with the default configurations
- Creating a Root Filesystem image
- Re-configuring the kernel under LTIB
- Selecting user-space packages
- Setup the bootloader arguments to use the exported RFS
- Debugging Uboot and the kernel by using Trace32
- Command line options
- Adding a new package
- Other deployment methods
- Creating a new package and integrating it into LTIB
- A lot of labs have been created to explain the usage of LTIB