Advanced Microcontrollers - ARM

Agenda

- STM32 SPI Programming
- Cortex-M DMA
- Bootloader
- ISP vs IAP
- STM32 Clock Configuration
- STM32 Reset

STM32 SPI Programming

- Refer yesterday's slides

Cortex-M DMA

- Direct memory access (DMA) is used in order to provide high-speed data transfer between peripherals and memory and between memory and memory.
- Data can be quickly moved by DMA without any CPU action.
- This keeps CPU resources free for other operations.
- STM32 User Manual (DMA)
    - DMA1 & DMA2 Memory interfacing Bus Matrix
    - Bus Matrix - Dual mode DMA
    - DMA controller Internals - 8 Req Streams
    - DMA1 Streams - Peripherals
    - DMA2 Streams - Peripherals
- DMA data flow
    - Source and Destination addresses
        - Memory to Peripheral: Memory base address is source and Peripheral register address is destination.
        - Peripheral to Memory: Peripheral register address is source and Memory base address is destination.
    - Width: Byte, Half-word, Word

    - Size: Number of data units
    - Pointer increment: Usually memory address is incremented after transfer of each unit of data.
    - Mode:
        - Normal mode is one time transfer (source to destination for given size).
        - In circular mode is continuous data transfer from source to destination. When pointer reach to end of memory address, it is incremented back to the starting address.
        - Double buffer mode used in Circular mode with two buffers.
    - Possible DMA configurations
        - P2M
        - M2P
        - M2M
        - P2P

Bootloader

- Embedded Bootloader
    - Bootloader program starts when processor/controller resets/Powered ON.
    - Typically BL is given by manufacturer and loaded into Boot ROM of the controller.
    - In STM32, BL program is in System memory (Boot ROM).
    - Useful to get the new program (Firmware) from user (via UART/USB/SPI/...) and burn it into flash ROM.
- Fixed memory map
    - Code area at 0x00000000
    - Data area at 0x20000000
- Boot modes defined by BOOT0 and BOOT1 pins.
    - BOOT1 (X) + BOOT0 (0) --> Flash memory
    - BOOT1 (0) + BOOT0 (1) --> System memory
    - BOOT1 (1) + BOOT0 (1) --> Embedded SRAM
- Physical Remap
    - 0x00000000 is remapped to boot memory as per boot mode configured
    - 0x00000000 --> Main flash when BOOT1 (X) + BOOT0 (0)
    - 0x00000000 --> System memory when BOOT1 (0) + BOOT0 (1)
    - 0x00000000 --> Embedded SRAM when BOOT1 (1) + BOOT0 (1)

- Embedded Bootloader
  - USART1
  - USART3
  - CAN2
  - USB OTG FS (Device Firmware Upgrade)
- Refer STM32 user manual and Application Note AN2606.

ISP vs IAP

- ICP (in-circuit programming)
  - ICP is the ability to program the Flash memory of a microcontroller using the JTAG protocol, the SWD protocol or the bootloader while the device is mounted on the user application board.
  - Also called as ISP (in-system programming)
- IAP (in-application programming)
  - IAP is the ability to reprogram the Flash memory of a microcontroller while the user program is running.

STM32 Clock Config

- STM32 User Manual (Chapter 6 & 7)
  - 7.2 & 7.3: Clock frequencies & clock configuration
- Clock sources
  - HSI - 16 MHz Internal RC Osc
  - HSE - External XTAL Osc -- STM32F407-DISC1 (8 MHz)
  - LSI - 32 KHz Internal RC Osc
  - LSE - External XTAL Osc -- STM32F407-DISC1 (32.768 KHz) -- Not fitted
- Default input clock (Fin) = HSI = 16 MHz
- Clock configuration
  - Fvco = (Fin / M) * N (100 to 432 MHz)
  - Fcclk = Fvco / P
  - Fusb = Fvco / Q
  - Fahb = Fcclk / H
  - Fapb1 = Fahb / apb1

    - Fapb2 = Fahb / apb2
- I2S (audio) clock
    - I2S bitrate = number of bits per channel × number of channels × sampling audio frequency

Illustration

- Fin = 8 MHz (Xtal)
- Fcclk = 168 MHz (Desired)
- Calculate P
    - P = Fvco / Fcclk
        - P = Fvco / 168
        - Fvco = 336
        - P = 2
- Calculate M
    - Fvcoin = (Fin / M)
        - Fin = 8 MHz
        - Fvcoin must (1 to 2 MHz)
        - M = 8
        - Fvcoin = 1 MHz
- Calculate N
    - Fvco = (Fin / M) * N
        - Fin = 8 MHz
        - M = 8
        - Fvco = 336 Mhz
        - N = 336
- Calculate Q
    - Fusb = Fvco / Q
        - Fvco = 336 Mhz
        - Fusb = 48 MHz
        - Q = 7
- Calculate H
    - Fahb = Fcclk / H

        - Fcclk = 168 MHz
        - Fahb = 168 Mhz (Max)
        - H = 1
- Calculate P1
    - Fapb1 = Fahb / P1
    - Fahb = 168 Mhz
    - Fapb1 = 42 Mhz (Max)
    - P1 = 4
- Calculate P2
    - Fapb2 = Fahb / P2
    - Fahb = 168 Mhz
    - Fapb2 = 84 Mhz (Max)
    - P2 = 2

STM32 reset

- STM32 user manual
    - 7.1: Reset
- System reset is generated when one of the following events occurs:
    - A low level on the NRST pin (external reset)
    - Window watchdog end of count condition (WWDG reset)
    - Independent watchdog end of count condition (IWDG reset)
    - A software reset (SW reset using RCC_CSR)
- System reset clears all registers (except Clock Control Regr (CSR) and backup registers).
- Power reset is generated when one of the following events occurs:
    - Power-on/power-down reset (POR/PDR reset) or brownout (BOR) reset
    - When exiting the Standby mode
- Power reset clears all registers (except backup registers).