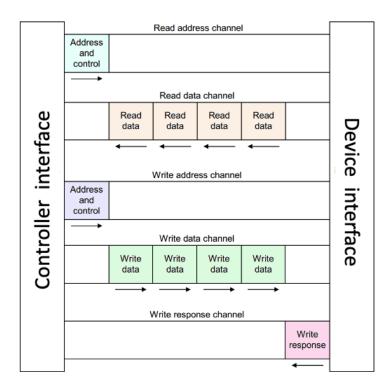
#### **AMBA (Advanced Microcontroller Bus Arch)**

- · Consists of 3 bus protocols
  - AXI (Advanced eXtensible Interface)
    - Connects high-speed components such as OoO CPUs and DRAM memory
  - Advanced High-performance Bus (AHB)
    - Connects lower speed (in-order) CPUs and SRAM memory
  - APB (Advanced Peripheral Bus)
    - Connects slow peripherals such as UART, Timer, GPIOs with microcontroller

#### **AXI**

- AXI separates different tasks into 5 channels
  - o Each task can happen independently. Burst wrtes doesn't affect slow reads.
  - Simple bus support only 1/2 channels
- AXI supports multiple outstanding transactions
  - o A controller can have multiple requests pending at the same time
    - multiple addresses being read from memory



#### **AMBA Channels**

#### APB / AHB : Two channels

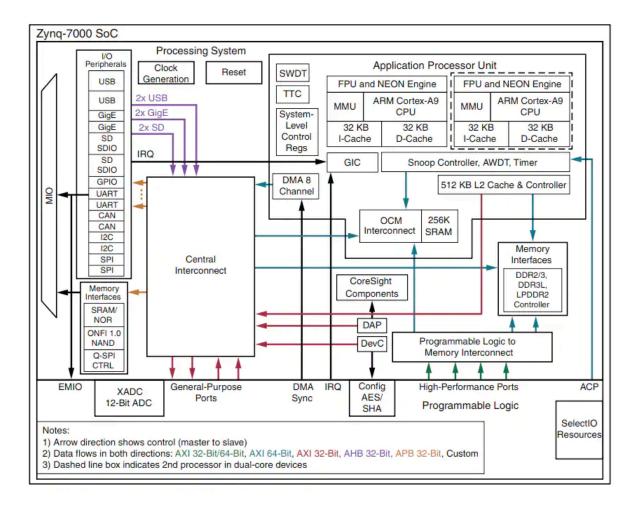
- APB:
  - Address
  - Data
- AHB:
  - Address/Data
  - Control
- Half-duplex (can only read or write at a time, but not both)

#### **AXI: Five channels**

- Write Address channel
- Write Data channel
- Write Response channel
- Read Address channel
- Read Data channel
- Full duplex (can read and write at the same time)

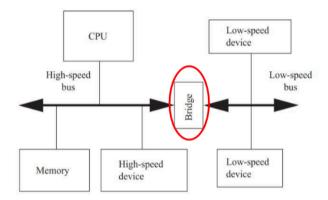
## **AXI vs. AHB vs. APB**

Feature	AXI	АНВ	АРВ
Data Width	32, 64, 128, or more bits	32 or 64 bits	Typically, 8, 16, or 32 bits
Data Transfer	Burst or single transfers	Burst or single transfers	Single transfers
Operation	Full duplex	Half duplex	Half duplex
Throughput	Very high	High	Low
Complexity	Complex	Moderate	Simple
Latency	Low latency	Medium latency	Higher latency
Efficiency	High	Moderate	Low
Power Consumption	High	Moderate	Very low
Controller support	Multiple controllers	Single controller	Single controller

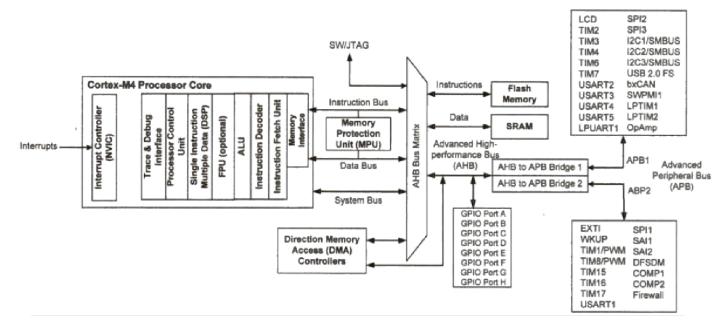


### **Multiple Buses**

- · Systems have different buses operating at different speeds, connecting different modules
- Need a bus interface/bridge to move data between them
- · High speed dev connected to high performance bus; low speed dev to different bus



#### **ARM Cortex M4**



#### STM32 - F4x Devices

2 AHB; 2 APB buses

CCM data RAM 64 KB External memory CLK. NE [3:0], A[23:0]. controller (FSMC) D[31:0], OEN, WEN, NJTRST, JTDI, AHB3 NBL[3:0], NL, NREG, JTAG & SW MPU SRAM, PSRAM, NOR Flash. JTCK/SWCLK JTDO/SWD, JTDO NWAIT/IORDY, CD PC Card (ATA), NAND Flash ETM NVIC NIORD, IOWR, INT[2:3] TRACECLK INTN, NIIS16 as AF TRACED[3:0] Arm Cortex-M4 168 MHz FPU Flash CACHE up to RNG Ethernet MAC DMA 1 MB MII or RMII as AF HSYNC, VSYNC MDIO as AF FIFO 10/100 Camera SRAM 112 KB PUIXCLK, D[13:0] interface USB DMA DP. DM SRAM 16 KB OTG HS ULPI:CK, D[7:0], DIR, STP, NXT USB 표 ID, VBUS, SOF DM OTG FS ID, VBUS, SOF DMA2 AHB2 168 MHz FIFC 8 Streams FIFO DMA1 Power managm VDD VDD = 1.8 to 3.6 V VCAP1, VCPA2 Supply RC HS PA[15:0] **GPIO PORT A** supervision RC LS VDDA, VSSA GPIO PORT B P L L 1 & 2 PVD **GPIO PORT C** PD[15:0] GPIO PORT D XTAL OSC OSC\_OUT PE[15:0] GPIO PORT E **IWDG** PF[15:0] GPIO PORT F control PWR VBAT = 1.65 to 3.6 V HCLKx ← **GPIO PORT G** @VB/ OSC32 IN PH[15:0] XTAL 32 kHz OSC32\_OUT GPIO PORT H S RTC PI[11:0] RTC AF1 GPIO PORT I Backup register S 4 KB BKPSRAM TIM2 出 TIM3 4 channels, ETR as AF 工 EXT IT. WKUP TIM4 DMA2 DMA1  $\Box$ TIM5 D[7:0] CMD, CK as AF SDIO / MMC AHB/APB2 AHB/APB1 TIM12 出 4 compl. channels (TIM1\_CH1[1:4]N, 4 channels (TIM1\_CH1[1:4]ETR, BKIN as AF TIM1 / PWM 16b TIM13  $\Box$ 4 compl. channels (TIM1\_CH1[1:4]N 4 channels (TIM1\_CH1[1:4]ETR BKIN as AF TIM14 16b TIM8 / PWM 16b USART2 RX, TX as AF CTS, RTS as AF 2 channels as AF TIM9 USART3 TIM10 16b 🕁 1 channel as AF UART4 RX, TX as AF WWDG TIM11 16b 😂 1 channel as AF UART5 RX, TX as AF MOSI/SD, MISO/SD\_ext, SCK/CK NSS/WS, MCK as AF RX, TX, CK, CTS, RTS as AF smcard USART1 MOSI/SD, MICK as AF
MOSI/SD, MISO/SD ext, SCK/CK
NSS/WS, MCK as AF SP2/I2S2 RX, TX, CK, CTS, RTS as AF smcard USART6  $\Leftrightarrow$ 16b 🔀 TIM6 SP3/I2S3 SPI1 TIM7 I2C1/SMBUS 16b SCL, SDA, SMBA as AF VDDREF\_ADC I2C2/SMBUS SCL, SDA, SMBA as AF  $\Xi$ 8 analog inputs commor to the 3 ADCs DAC1 ADC1 I2C3/SMBUS SCL, SDA, SMBA as AF ITF 8 analog inputs common to the ADC1 & 2 ΙF DAC<sub>2</sub> ADC2 bxCAN1 ADC3 bxCAN2 DAC1\_OUT DAC2 OUT MS19920V5

Figure 5. STM32F40xxx block diagram

1. The camera interface and ethernet are available only on STM32F407xx devices.

## **Why Multiple Buses**

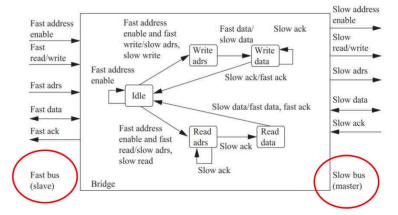
- More efficient use of the bigger bus
  - $\circ\;$  pack operations to and from multiple slow dev over a single operation of the large bus
- Simpler protocol for peripherals
  - o Connecting UART or timer over AXI adds a lot of complexity for no benefit

- Simpler address decoding logic
  - o Address decoder for the AHB bus can treat all APB peripherals as a single dev

#### **Bus Interface (Bridge)**

# Bridge is the logic that connects between buses

- A state machine or dedicated processor
- Also allows buses to operate independently
  - I/O operations may be done in parallel



Not always but, bridge is a device of the fast bus and controller of the slow bus.

- Takes commands from the fast bus
- Issues those commands to the slow bus, then returns the results from the slow bus to the fast bus
- It also serves as a protocol translator

Some components can be either Controller or Device at a time (hybrid)

 Example: a memory controller is a Device when being configured, but is a Controller when writing blocks to memory

## Multiple buses on bigger systems

