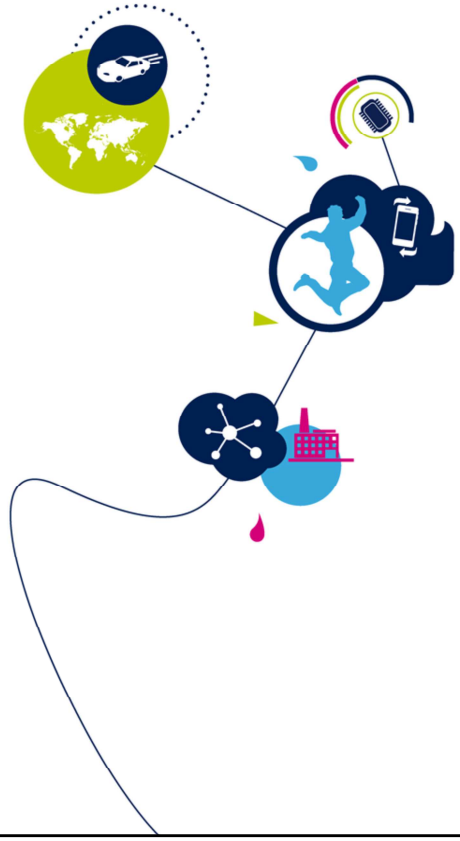
STM32H7 - AXI

Advanced eXtensible Interface (AXI) interconnect
Revision 1.0



life.augmented

Hello, and welcome to this presentation about the STM32H7 advanced extensible interface (AXI) interconnect.

Overview

- STM32H7 AXI interconnect features
  - 64-bit AXI bus switch matrix
  - AHB/AXI bridge function
  - Programmable traffic priority management
    - The quality of service (QoS) is software-configurable via the Global Programmer View (GPV)

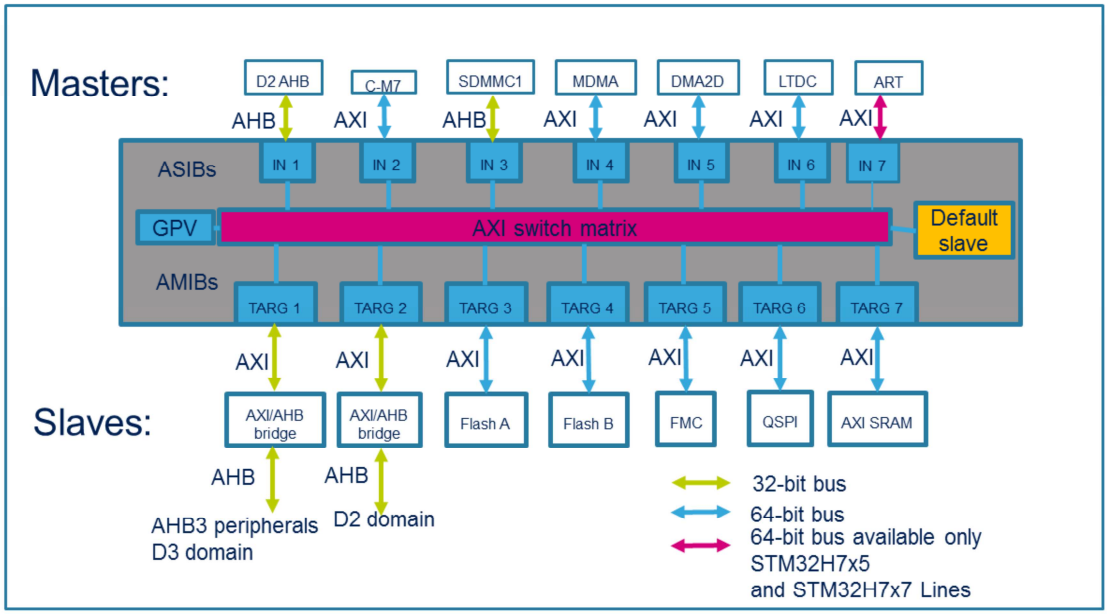
Application benefits

- Optimized data transfer bandwidth
- Configurable interconnect parameters
- Enhances real-time task responsiveness

life.augmented

The AXI (advanced extensible interface) interconnect is based on the ARM® CoreLink™NIC-400 Network Interconnect.
Its main features are :
- 64-bit AXI bus switch matrix with seven AMBA Slave Interface Blocks (ASIBs) and seven AMBA Master Interface Blocks (AMIBs), in D1 domain
- AHB/AXI bridge function built into the ASIBs
- concurrent connectivity of multiple ASIBs to multiple AMIBs
- programmable traffic priority management to ensure the quality of service (QoS)
- software-configurable via GPV

Block diagram



The STM32H7 AXI interconnect is shown in this figure:
The interconnect has seven initiator ports for the dual-core version, or ASIBs (AMBA slave interface blocks), and six initiator ports for the single-core version.
Both versions have seven target ports, or AMIBs (AMBA master interface blocks). The ASIBs are connected to the AMIBs via an AXI switch matrix.
Each ASIB is a slave on an AXI or AHB bus (advanced high-performance bus). Similarly, each AMIB is a master on an AXI or AHB bus. Where an ASIB or an AMIB is connected to an AHB bus, it converts between the AHB bus and the AXI bus protocol.
The AXI interconnect includes a global programmer view (GPV) which contains registers for configuring a few parameters, such as the quality of service (QoS) level at each ASIB.
Any access to unallocated address space is handled by

the default slave, which generates the return signals. This ensures that such transactions complete and do not block the issuing master and ASIB.

Key features (1/2)

- Priority-based arbitration when two AXI masters (ASIB) simultaneously attempt to access the same AXI slave (AMIB)
- AXI read channel and write channel priorities are independently configurable
  - Configurable priority values are from 0 (default) to 15 (highest priority)
  - If two coincident transactions arrive at the same AMIB, the higher priority transaction passes before the lower priority.
  - If the two transactions have the same QoS value, then a least recently-used (LRU) priority scheme is adopted.

life.augmented

The AXI switch matrix uses a priority-based arbitration when two ASIBs simultaneously attempt to access the same AMIB. Each ASIB has programmable read channel and write channel priorities, known as QoS, from 0 to 15, such that the higher the value, the higher the priority.
If two coincident transactions arrive at the same AMIB, the higher priority transaction passes before the lower priority. If the two transactions have the same QoS value, then a least recently-used (LRU) priority scheme is adopted.

Key features (2/2)

- QoS is useful for tasks such as graphics processing (LTDC, DMA2D)
- Priority assignment effect on full application should be studied carefully
  - Assigning a high priority to a master configured to make many and frequent accesses to the same slave can block access to that slave by other lower-priority masters
- Any access to unallocated address space is handled by the default slave, which generates the return signals
  - This ensures that such transactions complete and do not block the issuing master and ASIB

life.augmented

The QoS values should be programmed according to the latency requirements for the application. Setting a higher priority for an ASIB ensures a lower latency for transactions initiated by the associated bus master. This can be useful for real-time-constrained tasks, such as graphics processing (LTDC, DMA2D). Assigning a high priority to masters that can make many and frequent accesses to the same slave (such as the Cortex-M7 CPU) can block access to that slave by other lower-priority masters.