

## Lectures 18&19: Buses

- What is a bus?
- Bus Architectures
- Low speed peripheral buses
- AMBA

## Accessing memory locations from C

```
#define REG_FOO 0x40000140

unsigned int *reg = (unsigned int *) (REG_FOO);

main () {
    *reg += 3;
}
```

- What happens when **this** “instruction” executes?

### “\*reg += 3” is turned into a ld, add, str sequence

- Load instruction
  - A bus read operation commences
  - The CPU drives the address “reg” onto the address bus
  - The CPU indicated a read operation is in process
  - Some “handshaking” occurs
  - The target drives the contents of “reg” onto the data lines
  - The contents of “reg” is loaded into a CPU register
- Add instruction
  - An immediate add (e.g. add r0, r0, #3) adds three to this value
- Store instruction
  - A bus write operation commences
  - The CPU drives the address “reg” onto the address bus
  - The CPU indicated a write operation is in process
  - Some “handshaking” occurs
  - The CPU drives the contents of “r0” onto the data lines
  - The target stores the data value into address “reg”

### Details of the bus “handshaking” depend on the particular memory/peripherals involved

- SoC memory/peripherals
  - AMBA AHB/APB
- NAND Flash
  - Open NAND Flash Interface (ONFI)
- DDR SDRAM
  - JEDEC JESD79, JESD79-2F, etc.



# Interconnecting Components

- Need interconnections between
  - CPU, memory, I/O controllers
- **Bus:** shared communication channel
  - Parallel set of wires for data and synchronization of data transfer
  - Can become a bottleneck

# Bus Types

- Processor-Memory buses
  - Short, high speed
  - Design is matched to memory organization
- I/O buses
  - Longer, allowing multiple connections
  - Specified by standards for interoperability
  - Connect to processor-memory bus through a bridge

## Advanced Microcontroller Bus Architecture (AMBA)

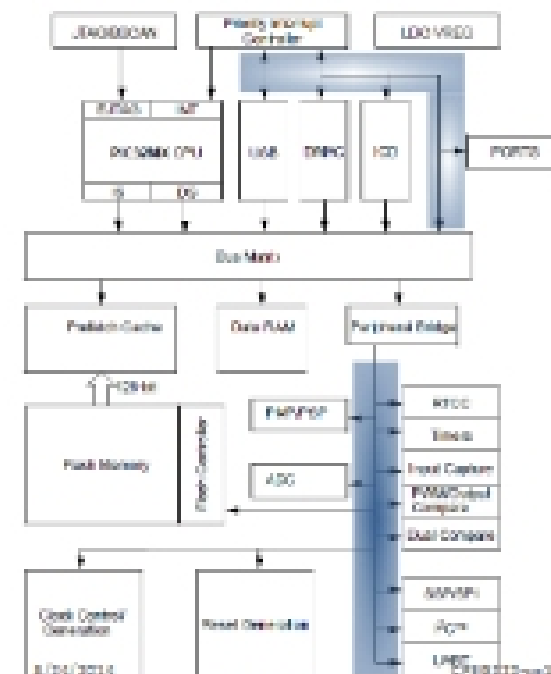
- Advanced High-performance Bus (AHB)
- Advanced Peripheral Bus (APB)



- AHB**
- High performance
  - Pipelined operation
  - Burst transfers
  - Multiple bus masters
  - Split transactions

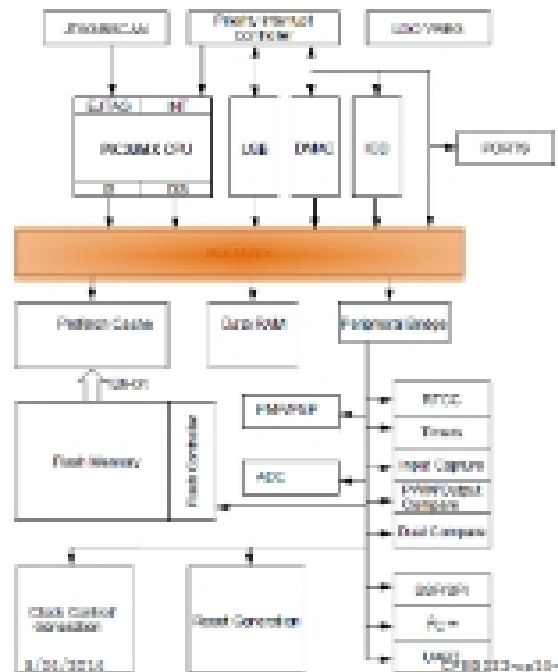
- APB**
- Low power
  - Latched address/control
  - Simple interface
  - Suitable of many peripherals

## Bussees on PIC32MX MCU



- Two peripheral buses:
- Main peripheral bridge connects most peripheral units to the bus matrix.
  - A high-speed peripheral bridge connects interrupt controller, DMA controller, in-circuit debugger, and USB peripherals.

# Busses on PIC32MX MCU



- Bus matrix is a switch that allows multiple accesses to occur concurrently in a system.
- It allows simultaneous accesses between different bus masters to different targets.
- It serializes accesses between different masters to the same target.
- The CPU is effectively two different bus masters:
  - I-side: instruction fetching
  - D-side: load and store
- Three other bus masters:
  - DMA controller
  - In-Circuit-Debugger Unit
  - USB controller

# Why have so many busses?

- Many designs considerations
  - Master vs Slave
  - Internal vs External
  - Bridged vs Flat
  - Memory vs Peripheral
  - Synchronous vs Asynchronous
  - High-speed vs low-speed
  - Serial vs Parallel
  - Single master vs multi master
  - Single layer vs multi layer
  - Multiplexed A/D vs demultiplexed A/D

# Bus Signals and Synchronization

- Data lines
  - Carry address and data
  - Multiplexed or separate
- Control lines
  - Indicate data type, synchronize transactions
- Synchronous
  - Uses a bus clock
- Asynchronous
  - Uses request/acknowledge control lines for handshaking

# Let us examine Low Speed Peripheral Units

- SPI
- I2C
- UART

