Architecture Design for Bus Interface

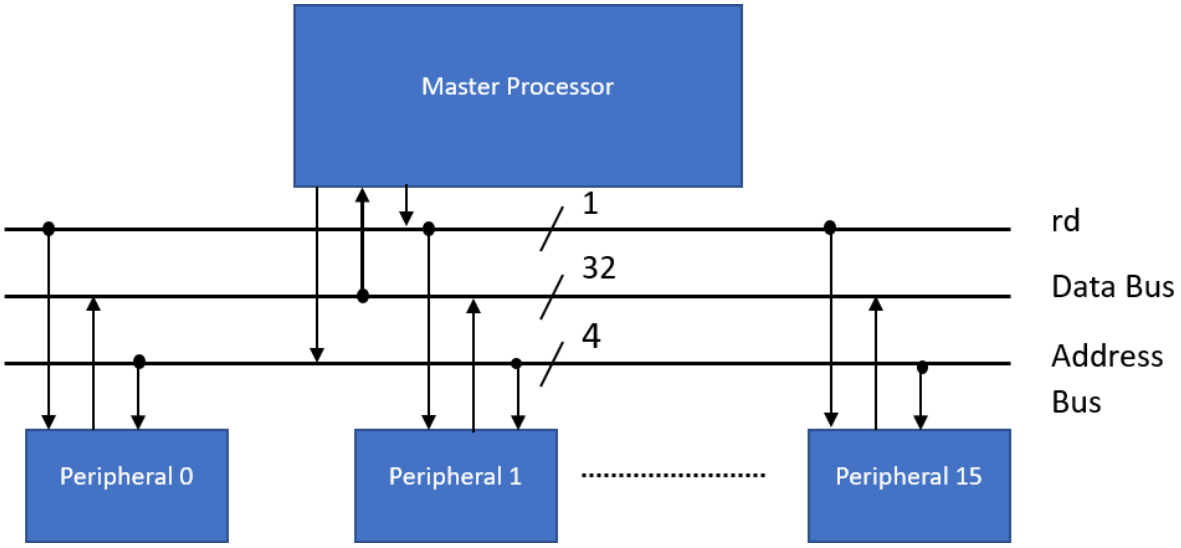
Master processor can read register from any peripheral.

- Each register has unique 4-bit address
- Assume 1 register/peripheral.
- Data is of 32 bit wide.
- rd is read enable flag;
  rd=1: loading of data from peripheral to Master Processor through Data Bus.
  Address for that data to be read is given through the 4 bit address bus which selects the particular register.
- There is total 16 peripherals each having unique 4 bit address.

Application of Bus Interface:

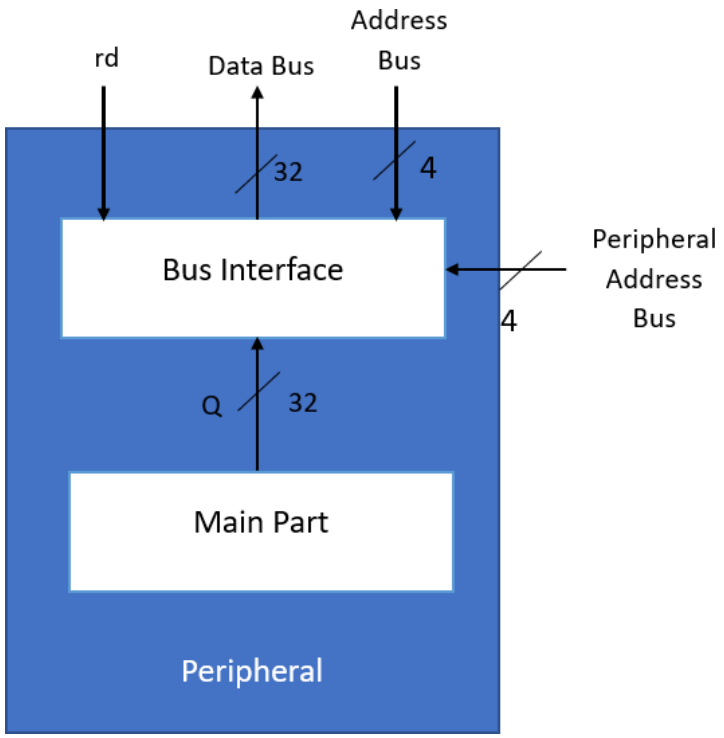
- It provides the interface to external memory and I/O devices via the System Bus.
- It performs various machine cycles such as memory read, I/O read, etc. to transfer data b/w memory and I/O devices.
- It is a main source of communication between two devices.

Macro Architecture:



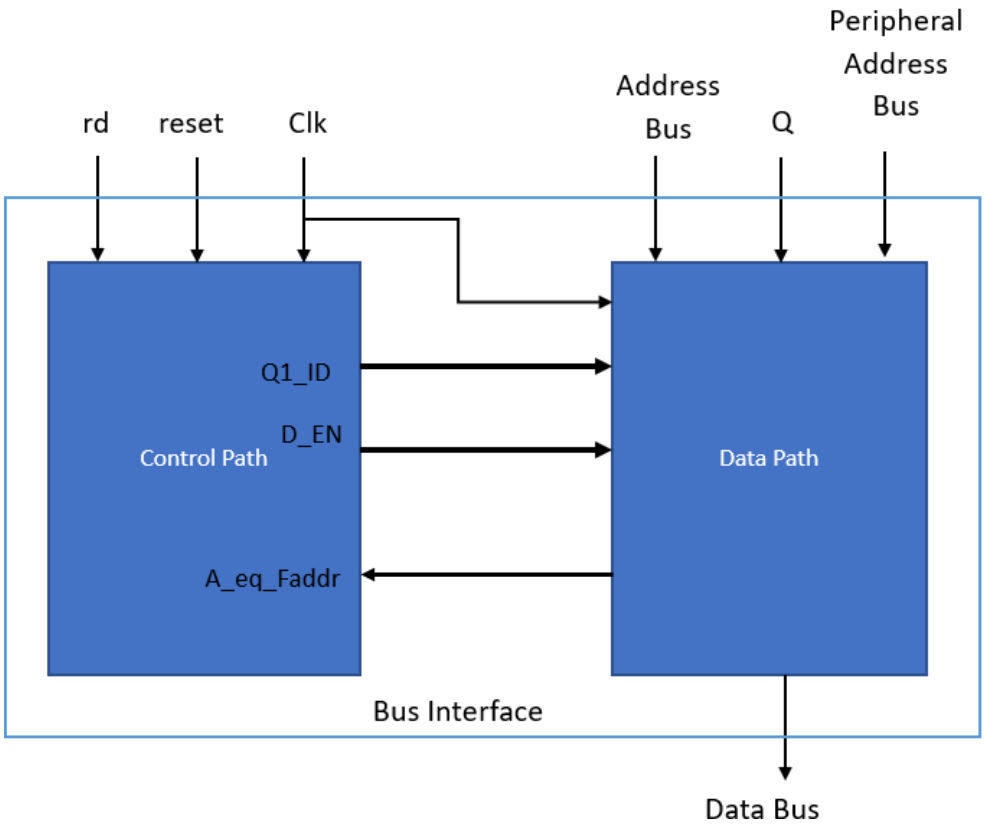
- The master processor is the main processor which can read register from any peripheral using a unique 4 bit address given to each register.
- Assuming 1 reg per 1 peripheral, the whole operation is conducted.
- Once the rd is enabled and address is provided & matched with the peripheral address, read cycle starts and the required data is taken from that particular location to the 32 bit data bus through a clocked register.
- The data is read per clock cycle (here at the positive edge)
- The 32 bit data is then provided to the master processor for further operations.

Micro Architecture:



- This is the inside structure of a peripheral device which consists of a Bus Interface and a main part.
- The main part comprises of a clocked register and a buffer which has a 32 bit O/P (Q).
- The register O/P Q is stored in a buffer at every clock cycle and once the D_en ==1, the data is given to the bus interface and reaches the processor through the Data bus.
- The bus interface is the source of communication b/w the master processor and the peripheral.
- INPUTS: one-bit rd, 4 bit address bus (A) and 4 bit peripheral address bus (Faddr)
- OUTPUTS: 32 bit Data Bus (D) for the data transfer.

Control Path:



Q1_ID: Local Reg O/P which is send to the Data Lines (D) through a clocked (positive edge triggered) register

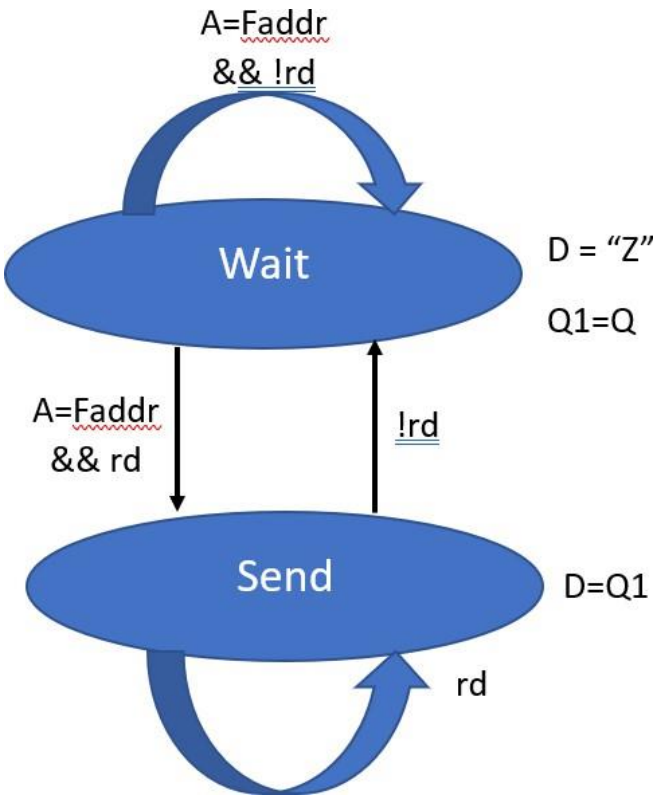
D_en: Enables the data bus for transferring the data to the data lines

A_eq_Faddr: It's a comparator circuit used for comparing the equality of the address on the add bus (A) and add on the peripheral bus (Faddr).

Reset: Resets the control path if enabled

Q: Peripheral O/P send to the data lines (D) through a clocked (positive edge triggered) register.

State Diagram:



INPUTS: rd (1bit), Q (32bits), Address Bus (A)- 4bits, Peripheral Add Bus (Faddr)- 4bits

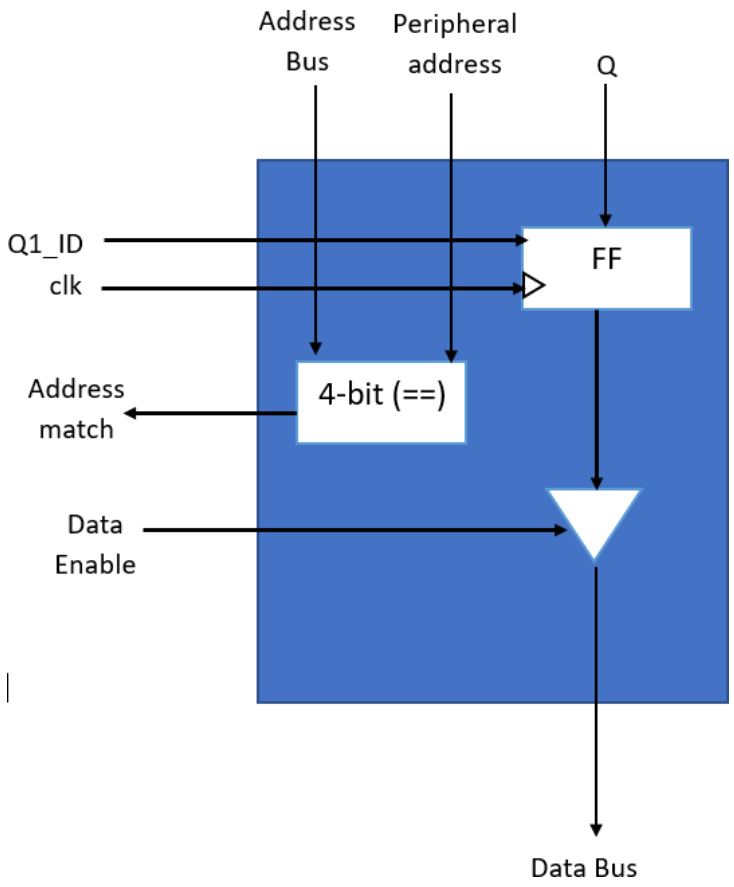
OUTPUT: Data bus (D) – 32bits

LOCAL REGISTER O/P: Q1 (32 bits)

STATE:

- Wait – Output = Z (Don't Care) condition and local reg o/p Q1= peripheral reg o/p Q1 until A=Faddr and rd==1
- Send – Once A=Faddr and rd==1, O/P Q1 moves into D and the process continues until rd==0.
- Once rd==0, the processor again moves into the wait condition as the processor is done reading the data lines.

Datapath:



- The address on the add bus and the add on the peripheral bus is first compared through a comparator.
- If equality matches, the add is sent to the master processor for further proceedings
- Q (peripheral O/P) and Q1_ID (local register O/P) is send to a buffer through a clocked register (in this case positive edge triggered).
- If the Data Enable (D_en) ==1, the data stored in the buffer is send to the data lines and loaded in the master processor through the 32 bit data bus.
- This is how a read cycle works in Bus Interface.