WISHBONE ARCHITECTURE

Wishbone interconnect makes SIC and design reuse easy by creating a standard exchange protocol

Wishbone Features:

Simple compact require few logic gates

Contains full set of popular bus transfer protocol

- R/W cycle
- BLOCK transfer cycle
- RMW cycle

- Modular Bus width
- Support BIG and LITTLE ENDIAN
- Handshake protocol
- Modular ADDR width.
- Master/Slave architecture
- Multi-MASTER capabilities

Objective:

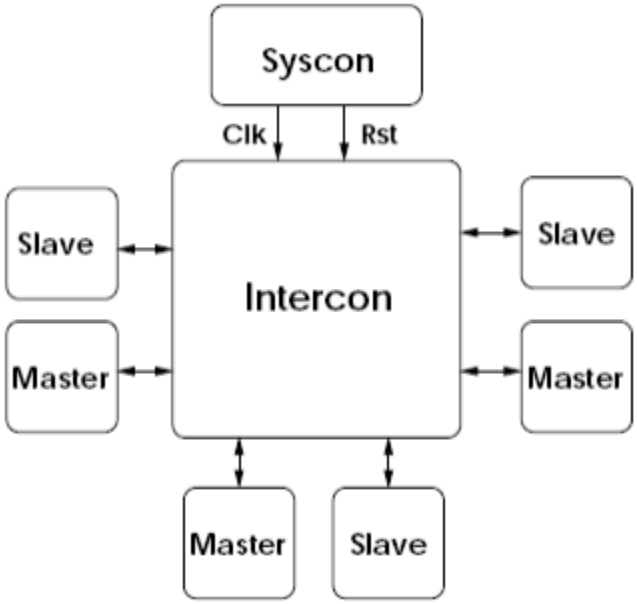
- Create flexible interconnection between semiconductor IP
- Enforce compatibility b/w IP cores

Advantage:

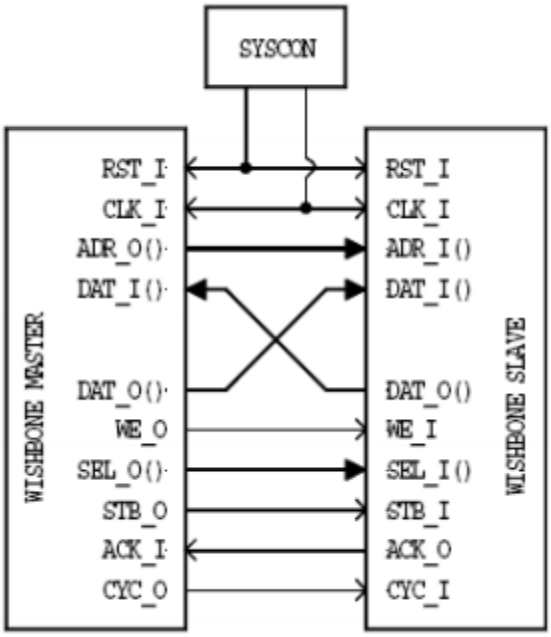
- Facilitates structural design methodologies
- Big projects can be broken to small parts and each team can build those small part and use wishbone architecture to communicate.
- Independent of underlying HDL language

- Portable interface independent of underlying semiconductor tech

WISHBONE BASIC:



- Master is an IP core which initiates a data transaction R/W/M
- Slave in turn responds to the data transaction initiated by the Master.
- INTERCON helps the data transfer
- SYSCON generates WISHBONE reset and clock signal
- Wishbone Interface signal



WISHBONE MASTER

RST_I	System reset input	Not Optional
CLK_I	Clock Input	Not Optional
ADR_O	Address output	Not Optional
DAT_I	Data input	Not optional
DAT_O	Data Output	Not optional
WE_O	Write enable output R= 0 W = 1	Not optional
SEL_O	Select output indicates where data is expected	YES optional
STB_O	Strobe output indicates a valid strobe	Not optional

ACK_I	Acknowledge Input when asserted indicates normal termination of a bus signal	Not optional
CYC_O	Cycle output When asserted shows a valid cycle in progress	Not optional

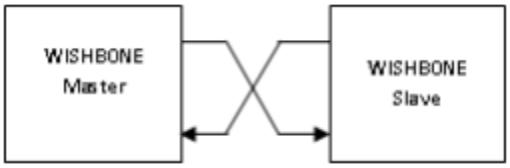
WISHBONE SLAVE

RST_I	System reset input	Not Optional
CLK_I	Clock Input	Not Optional
ADR_I	Address input	Not Optional
DAT_I	Data input	Not optional
DAT_O	Data Output	Not optional
WE_I	Write enable output R= 0 W = 1	yes optional
SEL_I	Select input indicates where data is expected	YES optional
STB_I	Indicates that Slave is selected when asserted slave responds to other wishbone signal	Not optional
ACK_O	Acknowledge Output when asserted indicates normal termination of a bus signal	Not optional
CYC_I	Cycle Input When asserted shows a valid bus cycle in progress	Not optional

WISHBONE Interconnection

1. **Point-to-Point**

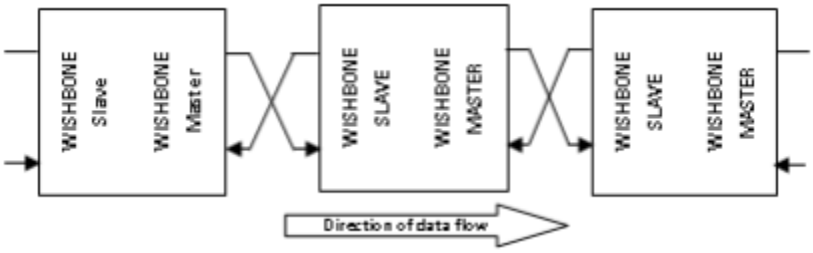
   Data transaction is controlled by handshake



2. Data Flow

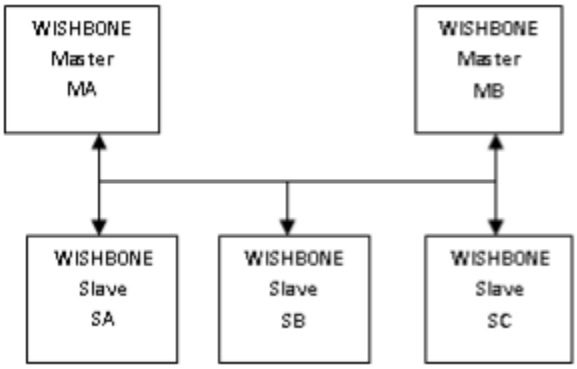
1. Process data in sequential manner

- Each IP core is contains both Master and a Slave Interface
- An IP core appears master to the next IP core but slave to the previous one.



3. Share Bus Interconnect

- Multi master connected to multi slave with a common bus.
- One master can get access to the bus a one time
- The traffic is controlled by an arbiter



4. **Cross Bar Switch Interconnect:**
    - Multi master and multi slave like share bus.
    - Multi master can communicate with the slave at the same time provided both the masters are not accessing the same slave.
    - Overall speed is higher than the share bus interconnect.

WISHBONE bus cycle

Three types

1. Single Read/Write
2. Block READ/WRITE
3. Read Modify Write

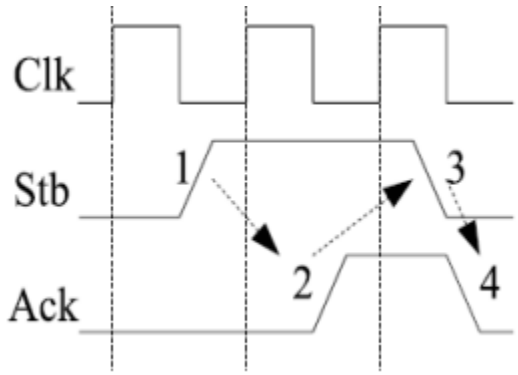
Handshaking protocol

Whole transfer process is classified into 4 parts:

1. Operation is requested
2. Slave is ready
3. Operation is over
4. Ready for a new operation

Handshake begins with Master asserting Strobe output indicating operation start

Strobe output is asserted till Slave asserts terminating output ack_o



Single Read/Write Cycle

READ:
Master starts the operation by presenting a

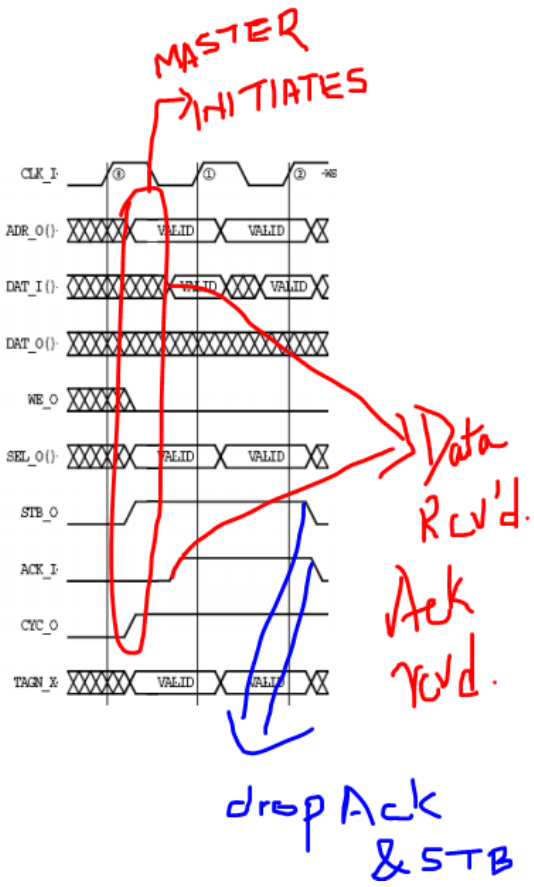
- valid address(addr_o)
- write enable signal 0
- Assert valid bank select indicated where data is expected
- Assert cycle output showing the starting of the cycle
- Assert strobe output to indicate transfer begins

Slave in response

- Presents valid data on data output
- Assert ackn signal to indicate the presence of valid data.

Master in response

- Latches the data in next clock cycle.
- Negate strobe and cycle output.



Block Transfer READ

Block transfer similar to single transfer

- Instead of dropping transfer (negating CYC_O after receiving the ack from slave) the transfer continues
- For every data transferred handshake is done.

- Wait Cycle can be put by master interface by pulling down strobe and ack signal.

Read-Modify_write(RMW) Cycle

- Used in Multi-processor and multi-tasking systems
- Use semaphores to share common resource
- Applied in disk controller, serial ports and memory

- Read portion (read Phase)
- Write Portion(write Phase)
- First phase of RMW cycle is checking and setting semaphore bits.

Designing WISHBONE compatible slave IP

- 16-bit WISHBONE slave with 8-bit granularity.
- 16-bit register can be accessed with either 8- or 16-bit bus cycles.
- SEL_I determine the selection of data byte
  SEL_I(1:0) : when sel[0] is asserted this means low byte is accessed while sel[1] is asserted higher byte is selected if both are asserted this means that 16 bit is selected.