Tools - Platform Designer (Formerly Qsys)

Notes are from the following video:
https://www.youtube.com/watch?v=Vw2_1pqa2h0

Automatic Interconnect Generation:

- Platform Designer automatically generates interconnects between the IPs you specify.

Options for Interconnect:

- Avalon Memory Mapped Interface (Avalon-MM)
  - Master components issue commands (read/writes) to addressable slaves.
  - These addressable slave components are often referred to as Avalon-MM slaves.
  - Supports simultaneous multi-mastering: Multiple masters can talk to multiple slaves all at once.
  - Avalon-MM supports a bunch of signal types:
    - Address bus
    - Data bus
    - Byte enable
    - read/wr control
    - … and more
    - BUT! Your custom Avalon-MM slave component only needs to support the signal types it needs.
- Avalon Streaming Interface (Avalon-ST)
  - Unidirectional: Source component -> Sink component only.
  - No memory map. Each streaming interface can have multiple channels instead
  - For use in situations where high-throughput and low latency is required. For example, Video/DSP data streams, etc.
- ARM AMBA AXI Standard Interface
  - Similar to Avalon-MM, but with more features:
    - Retains memory mapped master/slave interfaces
    - Separate addr/data/response channels
    - Out of order transactions
    - Secure transactions

Vocab

- Components: Your IP blocks that interact with the Platform Designer system and support one or more interconnect interfaces.
- Interface: The signals that describe component connections (clock, reset, src, sink, master, slave, etc.)
- Source: Avalon-ST sender
- Sink: Avalon-ST receiver
- Master: Avalon-MM or ARM AXI Interface which initiates transfers through commands
- Slave: Avalon-MM or ARM AXI interface which responds to commands
- Interrupt sender/receiver: Interfaces that send/receive IRQs.

- Any component you design can send/receive IRQs.

Interface Requirements

- Every component needs at least one clock interface (input or output or both).
- Every component must either generate or receive a reset signal.

Optional:

- IRQ receiver (32 separate interrupts, interrupt 0 is highest priority)

Avalon®-ST Interface Signals

Signal type	Width	Direction	Description
Fundamental signals			
ready	1	Sink → Source	Indicates the sink can accept data
valid	1	Source → Sink	Qualifies all source to sink signals
data	1-4096	Source → Sink	Payload of the information being transmitted
channel	1-128	Source → Sink	Channel number for data being transferred (if multiple channels supported)
error	1-255	Source → Sink	Bit mask marks errors affecting the data being transferred
Packet transfer signals			
startofpacket	1	Source → Sink	Marks the beginning of the packet
endofpacket	1	Source → Sink	Marks the end of the packet
empty	1-5	Source → Sink	Indicates the number of symbols that are empty during cycles that contain the end of a packet

- Again, when you create a custom component, you only need to implement the signals you want/need.

Avalon-ST Adapters:

- Added automatically during compile-time by Platform Designer.
- Handles semi-compatible interconnect between two components (different clock frequencies, different interface implementations, etc.).

Basic Avalon®-MM Master Interface Signals

Signal type	Width	Direction	Required	Description
address	1-64	Output	Y	*Byte* address corresponding to slave for transfer request; must align with data width *(discussed later)*
waitrequest waitrequest_n	1	Input	Y	Forces master to stall transfer until deasserted; other Avalon®-MM interface signals must be held constant
read read_n	1	Output	N	Indicates master issuing read request
readdata	8, 16, 32, 64, 128, 256, 512, 1024	Input	N	Data returned from read request
write write_n	1	Output	N	Indicates master issuing write request
writedata	8, 16, 32, 64, 128, 256, 512, 1024	Output	N	Data to be sent for write request
byteenable byteenable_n	2, 4, 8, 16, 32, 64, 128	Output	N	Specifies valid byte lane(s) for readdata or writedata (width = data width / 8)
lock	1	Output	N	Once master is granted access to shared slave, locks arbiter to master until deasserted
response	2	Input	N	Indicates successful transfer or not, as well as whether access is to undefined address location

Additional Master Interface Signals

Signal type	Width	Direction	Required	Description
Bursting transfers				
burstcount	1-11	Output	N	Indicates number of transfers in burst (# transfers = 2^(width – 1)); if bursts are reads, must include **readdatavalid** signal below
Pipelined transfers				
readdatavalid readdatavalid_n	1	Input	N	Indicates valid data from prior read transfer request is available on readdata input; only used with variable pipeline latency
writeresponsevalid	1	Input	N	Used along with **response** to indicate a write response one cycle (by default) after a write; increase with **minimumResponseLatency** property

- The only required signals are the address lines and the waitrequest signals

Basic Avalon®-MM Slave Interface Signals

Signal Type	Width	Direction	Required	Description
address	1-64	Input	N	*Word* address of slave for transfer request *(discussed later)*
waitrequest waitrequest_n	1	Output	N	Allows slave to stall transfer until deasserted; other Avalon®-MM interface signals must be held constant
read read_n	1	Input	N	Indicates slave should respond to read request
readdata	8, 16, 32, 64, 128, 256, 512, 1024	Output	N	Data provided to Platform Designer interconnect in response to read request
write write_n	1	Input	N	Indicates slave should respond to write request
writedata	8, 16, 32, 64, 128, 256, 512, 1024	Input	N	Data from the Platform Designer interconnect for a write request
byteenable byteenable_n	2, 4, 8, 16, 32, 64, 128	Input	N	Specifies valid byte lane for readdata or writedata (width = data width / 8)
begintransfer begintransfer_n	1	Input	N	Asserts at the beginning (first cycle) of any transfer
response	2	Output	N	Indicates successful transfer or not, as well as whether access is to undefined address location; interconnect provides OK if no slave response

Additional Slave Interface Signals

Signal type	Width	Direction	Required	Description
Bursting transfers				
burstcount	1-11	Input	N	Indicates number of transfers in burst (# transfers = 2^(width – 1)
Pipelined transfers				
readdatavalid readdatavalid_n	1	Output	N	Indicates valid data from prior read transfer request is available on readdata input; only used with variable pipeline latency
writeresponsevalid	1	Input	N	Used along with **response** to indicate a write response one cycle (by default) after a write; increase with **minimumResponseLatency** property

See the Avalon® specification for examples and detailed timing diagrams for these advanced transfer types

- No signals are required.

Slide summarizing the mandatory signals:

Avalon®-MM Interface Minimum Signal Requirements

Master interface

- Must have **address** and **waitrequest**
- To execute write transfers
  - **write**, **writedata**
- To execute read transfers
  - **read**, **readdata**

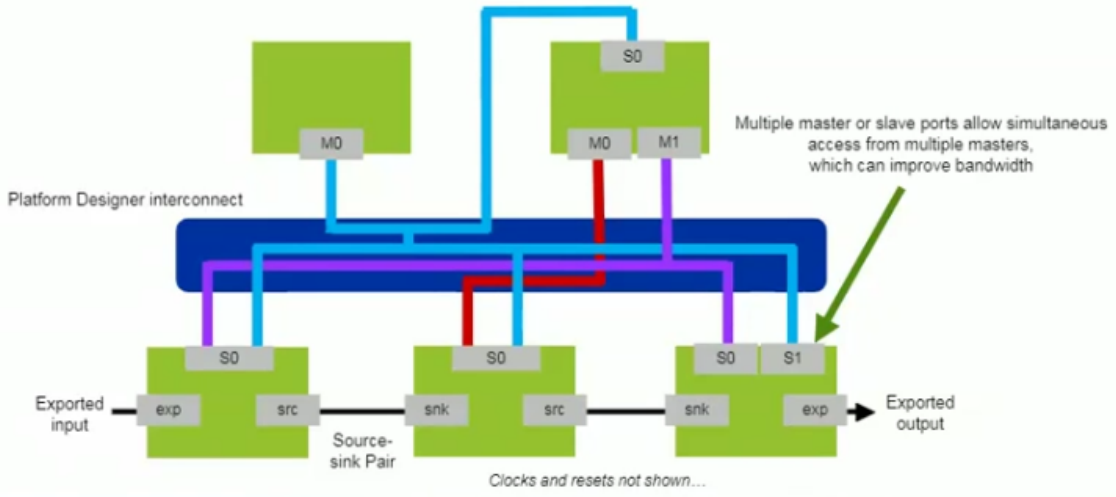
Slave interface

- No **address** or **waitrequest** necessary; single address location set in Platform Designer
- For a writable slave
  - **write**, **writedata**
- For a readable slave
  - **readdata**

- If you do not use an address bus, the slave can be accessed through an address location set in Platform Designer.

Example system:

Connections Through Avalon®-ST or -MM Interfaces



- Notice how you can have multiple master or slave ports.

Conduit Interfaces:

- Use them for reaching out to external peripherals.
- Try not to use them for internal signals, since Platform Designer cannot verify them or auto-generate Adapters.