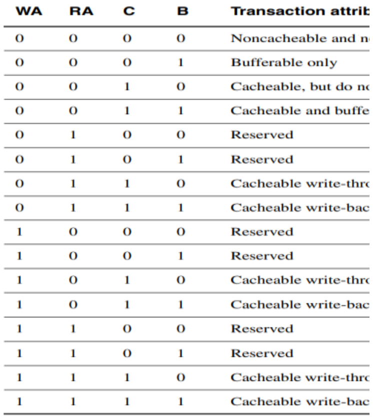
1	Power on Rst	Reset the entire design
2.2.1	Master Write(Bus model **master**)	Master writes,Check if AWvalid goes high and if AWADDR is correct when it supposed to and awaddr is outputed correctly for addressing phase, check Wdata,Wstrb, Wlast,Wvalid and finally check Bready. (AWLEN = 4'b0 Burst lenth is 1) (AWSIZE = 4'b0 each transaction is 1 byte)
2.2.2	Master Read(Bus model **Master**)	Check ARADRR is correct and ARVALID goes high at a resonable time period after (for addressing phase). Check RRVALID goes high an appropriate time (ARLEN = 4'b0 Burst lenth is 1) (ARSIZE = 4'b0 each transaction is 1 byte)
2.2.3	**AWLEN and AWREN > 0** Read and write Random burst lengths(diffrent number of transfers)(Bus model **Master**)	Random burst length and make sure AWVALID,AWADDR,ARADDR, ARVALID,WDATA,WSTRB,WLAST,BREADY,ARADDR,ARVALID,RREADY is correct
2.2.4	AWSIZE and ARSIZE > 0 read and write random data sizes(different data sizes) (BUSmodel **master**)	Random burst sizes and make sure AWVALID,AWADDR,ARADDR, ARVALID,WDATA,WSTRB,WLAST,BREADY,ARADDR,ARVALID,RREADY is correct
2.2.5	AWBURST and ARBURST all types, fixed, Incr,wrap, reseved(BUSmodel **master**)	Main check is WDATA correct, WLAST going high correctly
2.2.6	AWLOCK|| ARLOCK == EXCLUSIVE Access (semaphore accesses) (bus model **master**)	**I actually do not think you can check anything here it is all a master thing**
2.2.7	AWCACHE and ARCACHE all combinations (bus model **MASTER**)	Check all possible combinations, going to be intresting to check (**ASK ABOUT BITS 2-3)**
2.1.8	ARPROT and AWPROT Protection	Check all possible combinations, privileged vs unprevliged,secure vs unsecure,data vs instruction(**Still confused on what this entails)**

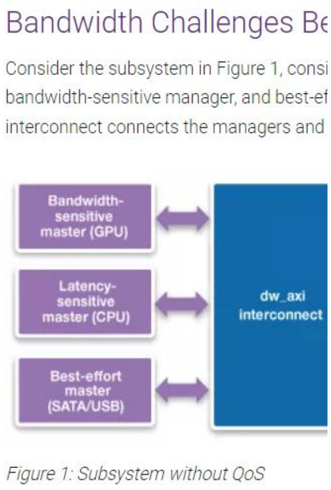
2.2.9	AWQOS and ARQOS	Change the QOS on the bus and check if the correct slave is being serviced, for example if memory is being accessed more by one master than the other master it should have a higher QOS and that transaction should take precedence if the master interface has more than one option
2.2.10	AWREGION and ARREGION	For a multimodel slave that has registers,data buffers, logic ETC communicate different regions to help with decoding Check if slave recieves data in the correct region and W ready goes high appropriately
2.2.11	ARUSER AWUSER BUSER RUSER	User defined signal only custom info I can think to test is like power dissaption or something wacky like that, this is completely up to the designer
2.2.12	ARID AWID, RID, WID, BID standard check	Check if ID's match when doing trasncations on change ID up and make sure data is not read for example change the Master ID after reading
2.2.13	Unalligned transfers	Start the AW or AR addr at a weird spot so not the entire data is received (example: 0x1, size is 32 bits, on the 1st transfer only 1-3 bytes are written or read and then from there each word should be fully seen (page 54 small manual
2.3.1	**ERROR** Send an unsupported adress	Send incorrect read and write address and make sure SLVERR is being received
2.3.2	**ERROR** timing violtaion	Timing violation, changing values at posedge
2.3.3	**ERROR** Unalligned data transfers, sending more data than in a transaction	Should see a SLVERR since to much data is being sent
2.3.4	**ERROR** Give an undecoable address for read and right	Should see a DECERR since to much data is being sent

		1	100	
		1	100	
		1	100	
		1	100	
		1	100	
		1	100	
		1	100	
		1	100	
		1	100	



WA	RA	C	B	Transaction attrib
0	0	0	0	Noncacheable and n
0	0	0	1	Bufferable only
0	0	1	0	Cacheable, but do n
0	0	1	1	Cacheable and buffe
0	1	0	0	Reserved
0	1	0	1	Reserved
0	1	1	0	Cacheable write-thr
0	1	1	1	Cacheable write-bac
1	0	0	0	Reserved
1	0	0	1	Reserved
1	0	1	0	Cacheable write-thr
1	0	1	1	Cacheable write-bac
1	1	0	0	Reserved
1	1	0	1	Reserved
1	1	1	0	Cacheable write-thr
1	1	1	1	Cacheable write-bac

		1	100	Quality of Service is priotity
		1	100	**ASK REGION**
		1	100	
		1	100	
		1	100	
		1	100	
		1	100	
		1	100	
		1	100	
		1	100	



Bandwidth Challenges Be

Consider the subsystem in Figure 1, consi
bandwidth-sensitive manager, and best-ef
interconnect connects the managers and

Bandwidth-sensitive master (GPU)
Latency-sensitive master (CPU)
Best-effort master (SATA/USB)
dw_axi interconnect

Figure 1: Subsystem without QoS

utes
onbufferable
ot allocate
erable, but do not allocate
ough, allocate on reads only
k, allocate on reads only
ough, allocate on writes only
k, allocate on writes only
ough, allocate on both reads and writes
k, allocate on both reads and writes

etween Managers and Subordinates

isting of different types of managers (latency-sensitive manager, ffort manager) and a shared-memory subordinate. The DW_axi subordinate.