DØ Level 2 Magic Bus

Drew Baden, Bob Hirosky, Reinhard Schwienhorst
Updated and annotated from the original documents of M. Campbell, C. Murphy, D.Baden

9/28/2004

The most current version of this note may be found at the L2βeta web site:
http://galileo.phys.virginia.edu/~rjh2j/l2beta#specs

The Level 2 Magic bus (MB) is a 128-bit wide data bus used for communication between modules in the Level 2 Trigger Processor crate. The MB is mounted on the P3 backplane of the DØ VME crate between slots 7 and 20. The connections to the MB are made through an AMP 235 pin, 2mm auxiliary connector. The list of signals is given in Table 1.

Programmed I/O transactions

A module initiating a data transfer must first gain bus mastership. The arbitration priority is determined by the position within the VME crate, with the **lowest** numbered slots having the **highest priority**. A module requesting bus mastership asserts BOSSREQ if there is no current master, i.e. BOSS is not asserted. BOSSREQ is connected to BOSSGRIN at the highest priority slot (slot 7 in the DØ Level 2 crates). When BOSSGRIN goes true the state of the internal request is latched. Then either BOSSGROUT is asserted and the grant is passed to the next module, or BOSS is asserted and the current module becomes bus master. During the time the current module is master, BOSS is asserted which prevents BOSSREQ from going true. A module releases BOSS only when it has completed all transactions. At the lowest priority slot (slot 19 in the DØ Level 2 crates) BOSSGROUT is connected to BOSS to prevent dead locks. Slots that are empty or contain modules that cannot be MB masters must have BOSSGRIN jumpered to BOSSGROUT.

A diagram of the MBus arbitration logic is shown in Fig. 1. LOCAL_REQ represents a request for bus mastership from within a processor card. As stated above this request will not be placed on the MBus until the bus is free (BOSS is de-asserted). LOCAL_BOSS is simply a line internal to the processor board reflecting its status as bus master. A timing diagram of the bus arbitration sequence is shown in Fig. 2.

Figure 1. Magic Bus arbitration logic (see p. 9 for an update to this logic).

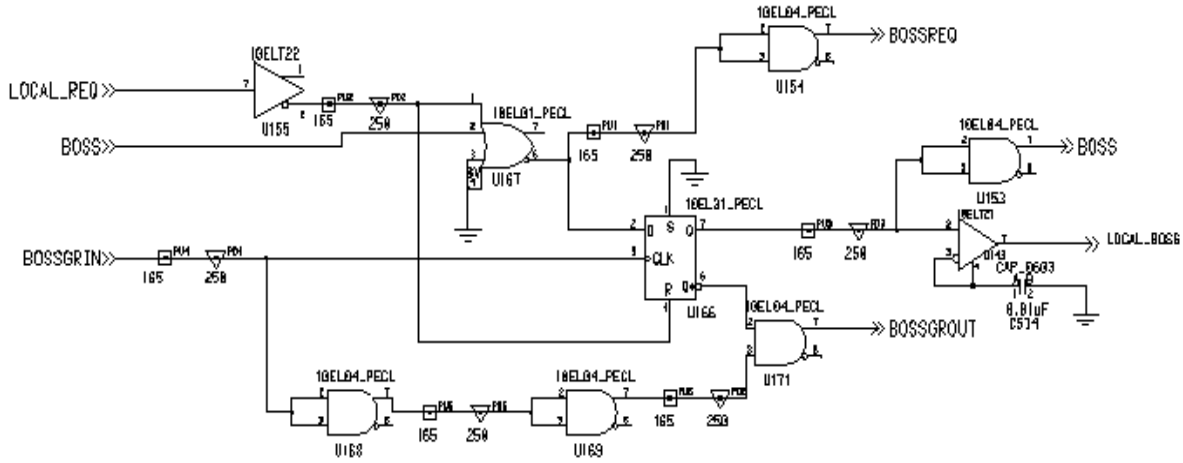
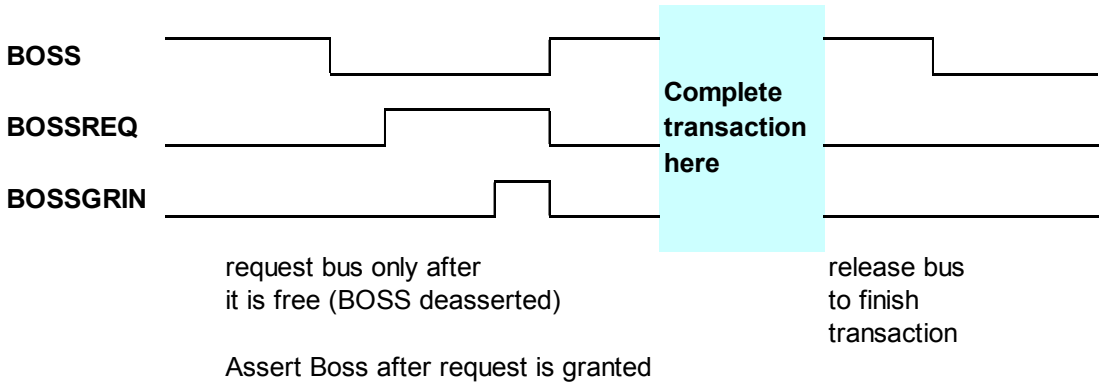


Figure 2. Magic Bus mastership arbitration



After a module becomes master, it may initiate one or more data transactions. The master asserts the address of the source of data and the RD/WR* strobe high if it is a read cycle, or the destination address and data if a write. **After waiting at least 10 nanoseconds, the master asserts DSTROBE***. A slave module, seeing DSTROBE* go active, examines the MBAD lines. If the slave recognizes the address, it either puts the

corresponding data on MBDATA if a read, or latches the data if a write, and **after a minimum of 10ns asserts DDONE***. The master upon seeing DDONE*, latches the data if a read, and then removes DSTROBE* and the slaves remove their data and strobes.

Modules may delay removal of the strobes or DDONE* in order to prevent another cycle from being started if they have not finished processing the current cycle. The master may not remove BOSS if DSTROBE* or DDONE* is asserted. In order to prevent a lockup condition in the master (for example if it tries to access a MBus address where no targets are programmed to respond), the transaction can be halted after a timeout period. *We recommend setting this timeout period to 1μs (i.e. halt the transaction if DDONE* is not received with this time period). No further action would be taken in the case of a write, a value of 0 could be returned in the case of a read.* Figures 3-4 show timing diagrams for Write and Read transactions.

Figure 3. Magic Bus write transaction

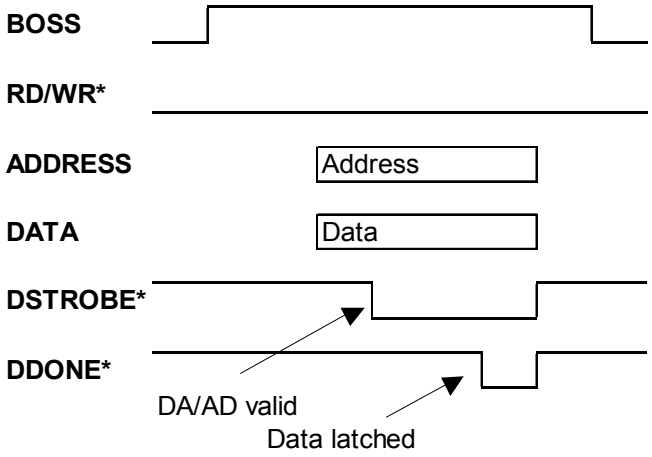
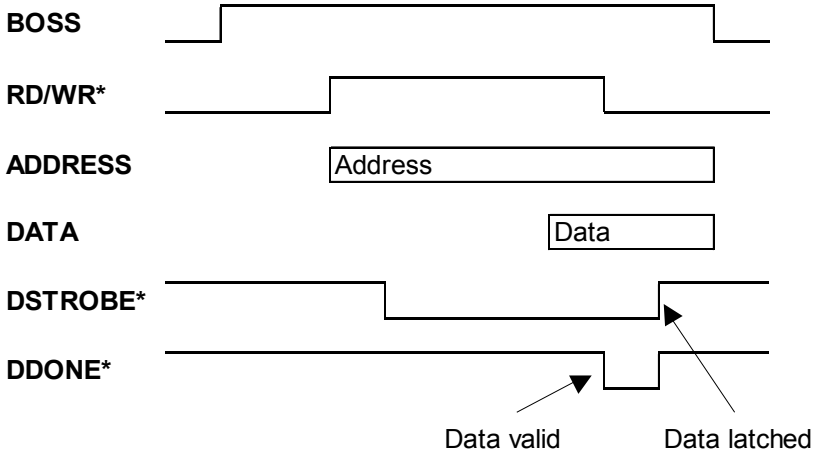


Figure 4. Magic Bus read transaction



MBRESET* will cause all slaves to remove all data and strobes from the backplane, as well as to reset appropriate internal registers.

Data broadcast transactions

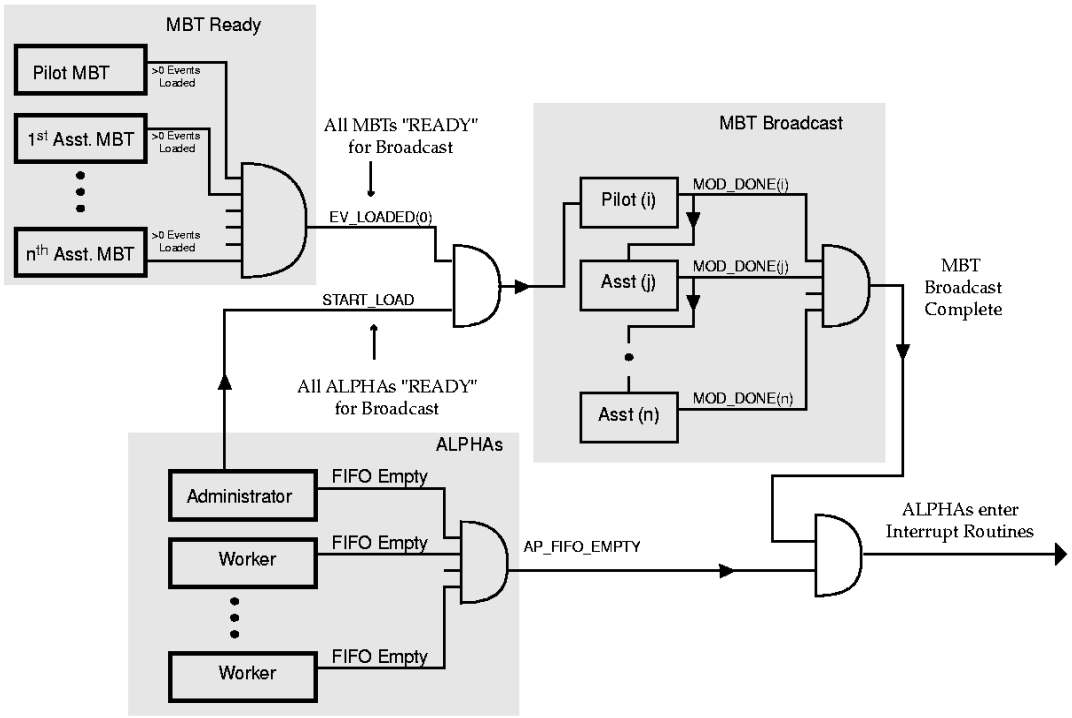
Magic Bus addresses 0-1023 (lower 10 bits) are reserved for broadcast-type writes. Other than these restrictions, however, addresses are assigned to modules by any agreed upon convention. A module may use as many addresses as necessary. An address specifies both the module and the module's internal address. Processor cards are always targets for broadcast writes.

The following is a DØ-specific description of the broadcast sequence. See section 4 of the MBT TDR[1] for more discussion of event broadcasting in the DØ crates. The special signals on the MBUS used in broadcasting are START_LOAD*, BUFFER (1:0), EV_LOADED(3:0), MOD_DONE(19:0), DONE_OUT and AP_FIFO_EMPTY. EV_LOADED(3:0) is an open collector signal. *Only EV_LOADED(0) is used in the DØ broadcast sequence, the BUFFER(1:0) lines are unused.* When an event's data has been transferred into the MBT's, EV_LOADED(0) is aserted. The actual broadcasting of data begins only after the crate Administrator processor card issues a START_LOAD* command. START_LOAD* informs MBT's that they can begin to load the next event into the level 2 processors. Note that START_LOAD* *may* be generated before all of the MBT cards have received data and broadcasts will begin when EV_LOADED is satisfied. MOD_DONE is a bus of signals from each slot in the crate saying that that slot is finished sending event data to the processors. Each slot also has a DONE_OUT line which is routed on the backplane to the appropriate MOD_DONE pin. (MAPPING OF DONE_OUT PINS TO MOD DONE LINES?)

AP_FIFO_EMPTY is an open-collector signal that is driven by the processor boards. When a processor board is still moving data from the event fifo to main memory, it drives AP_FIFO_EMPTY low. When all of the data from an event has been moved to main memory, the processor board releases AP_FIFO_EMPTY, which will then go high only when all processors have empty FIFO's. The signals MOD_DONE(19:0) and AP_FIFO_EMPTY notifies the processors that the current event can begin processing and the next event can begin loading. The broadcast sequence is illustrated in Fig. 5. The sequence begins by the Administrator processor sending a START_LOAD signal. When an event is fully loaded, the MBTs start broadcasting data beginning with the pilot MBT in the highest priority slot and ending with the MBT in the lowest priority slot. A timing diagram for the broadcast is given in Figure 6 (from the MBT TDR).

[1] A link to the MBT TDR is available at http://galileo.phys.virginia.edu/~rjh2j/l2beta#alpha_docs

Figure 5. MBT to L2 processor broadcast sequence



Miscellaneous electrical details

All bussed signals on the backplane are terminated at each end with a 220 ohm resistor to +5V and a 330 ohm resistor to ground. The drivers for backplane signals should be capable of sinking 64 mA of current, such as the 74FCT244 or the 74FCT245.
The 4 bus arbitration lines: BOSS, BOSSGRIN, BOSSGROUT, and BOSSREQ are PECL logic level signals. The schematic below gives the standard arbitration circuit to be used for each board. The two signals, LOCAL_REQ and LOCAL_BOSS in the schematic are TTL level and are used to interface with the board's control logic. Each PECL line has a 165 ohm pull up resistor to +5V and a 250 ohm pull down resistor to ground.

Figure 6. Timing diagram for a broadcast cycle.

MBT_DONE = all MBT MOD_DONE lines asserted (all MBTs finished broadcasting)
AP_FIFO_EMPTY = all ALPHA DMA FIFOs empty (drained of broadcasted data)
ALPHA_REC_DONE = MBT_DONE.and.AP_FIFO_EMPTY (all ALPHAs enter interrupt routines)
Pilot_ENABLE = START_LOAD.and.EV_LOADED(0)
Assistant_ENABLE = START_LOAD.and.MOD_DONE(previous)

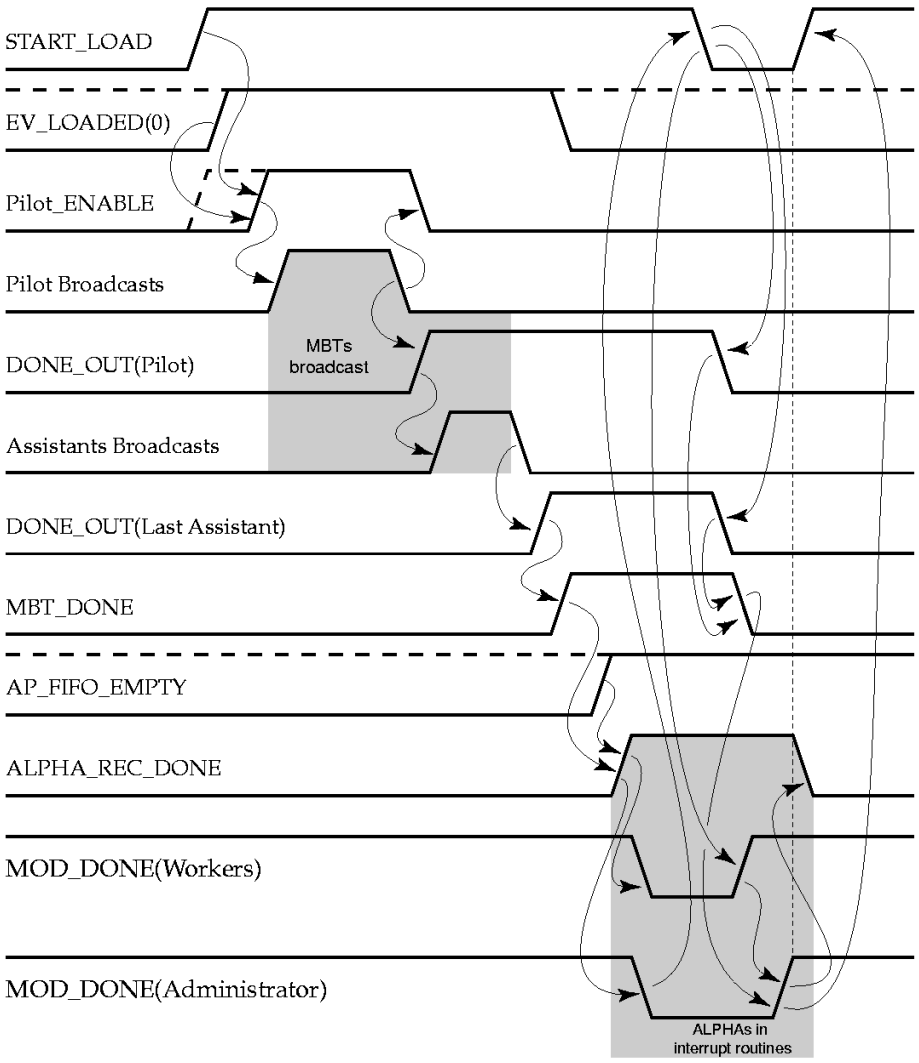


Table 1. Magic Bus signal descriptions.

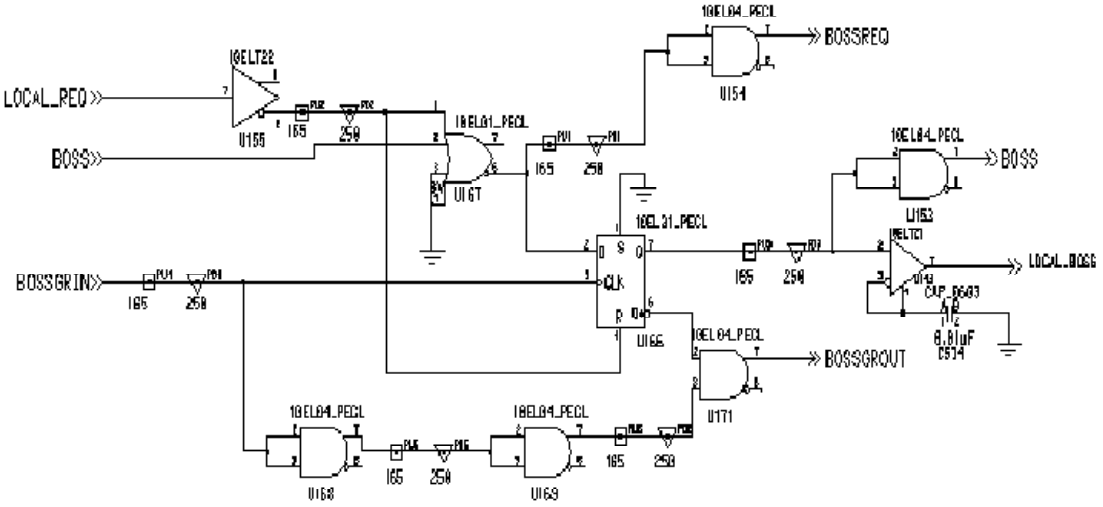
Type	Bits	Logic	Mnemonic	Description
Data	128	TTL	MBDATA	Data lines
Address	32	TTL	MBAD	Address lines
Cycle Type	1	TTL	RD/WR*	Asserted for read cycle, otherwise a write
Timing	1	TTL	DSTROBE*	For a read, address is valid; for a write, address and data are valid.
	1	TTL	DDONE*	For a read, data is valid; for a write, address and data are valid.
Arbitration	1	PECL	BOSS	A module is in control of the bus.
	1	PECL	BOSSREQ	A module requests control of the bus.
	1	PECL	BOSSGRIN	A module may take control of the bus.
	1	PECL	BOSSGROUT	The next module may control the bus.
Special	1	TTL	MBRESET*	Reset MB backplane- remove all strobes and data.
	1	TTL	START_LOAD*	MBT may start loading event data into the processors
	2	TTL	BUFFER(1:0)	Buffer number of event to start loading
	4	TTL	EV_LOADED(3:0)	Open collector signal driven by input modules when all data for an event has been read into the interface card
	1	TTL	AP_FIFO_EMPTY	Open collector signal driven by alpha processors when all event data has been transferred from the FIFO to main memory
	19	TTL	MOD_DONE(18:0)	Done signals from each level 2 interface module. L2 processors can see when all modules have finished sending buffer data.
	1	TTL	DONE_OUT	Each Module has a DONE_OUT signal that is routed on the backplane to the appropriate MOD_DONE bit.
Ground	18		GND	Signal return

MOD_DONE: In each VME slot DONE_OUT maps to the MOD_DONE bus as follows: MOD_DONE(n) = DONE_OUT(slot n-3). Therefore, the DONE_OUT line for slot 7 maps to MOD_DONE(4), DONE_OUT(slot 8) maps to MOD_DONE(5), etc. MOD_DONE(3:0) are not connected to any DONE_OUT pins in the DØ crates.

Table 2. DØ Magic Bus backplane signals

PIN	A	B	C	D	E
1	GND	GND	GND	GND	GND
2	MBDATA(0)	MBDATA(1)	MBDATA(2)	MBDATA(3)	MBDATA(4)
3	MBDATA(5)	MBDATA(6)	MBDATA(7)	MBDATA(8)	MBDATA(9)
4	MBDATA(10)	MBDATA(11)	MBDATA(12)	MBDATA(13)	MBDATA(14)
5	MBDATA(15)	MBDATA(16)	MBDATA(17)	MBDATA(18)	MBDATA(19)
6	MBDATA(20)	MBDATA(21)	GND	MBDATA(22)	MBDATA(23)
7	MBDATA(24)	MBDATA(25)	MBDATA(26)	MBDATA(27)	MBDATA(28)
8	MBDATA(29)	MBDATA(30)	MBDATA(31)	MBDATA(32)	MBDATA(33)
9	MBDATA(34)	MBDATA(35)	MBDATA(36)	MBDATA(37)	MBDATA(38)
10	MBDATA(39)	MBDATA(40)	MBDATA(41)	MBDATA(42)	MBDATA(43)
11	MBDATA(44)	MBDATA(45)	GND	MBDATA(46)	MBDATA(47)
12	MBDATA(48)	MBDATA(49)	MBDATA(50)	MBDATA(51)	MBDATA(52)
13	MBDATA(53)	MBDATA(54)	MBDATA(55)	MBDATA(56)	MBDATA(57)
14	MBDATA(58)	MBDATA(59)	MBDATA(60)	MBDATA(61)	MBDATA(62)
15	MBDATA(63)	MBDATA(64)	MBDATA(65)	MBDATA(66)	MBDATA(67)
16	MBDATA(68)	MBDATA(69)	GND	MBDATA(70)	MBDATA(71)
17	MBDATA(72)	MBDATA(73)	MBDATA(74)	MBDATA(75)	MBDATA(76)
18	MBDATA(77)	MBDATA(78)	MBDATA(79)	MBDATA(80)	MBDATA(81)
19	MBDATA(82)	MBDATA(83)	MBDATA(84)	MBDATA(85)	MBDATA(86)
20	MBDATA(87)	MBDATA(88)	MBDATA(89)	MBDATA(90)	MBDATA(91)
21	MBDATA(92)	MBDATA(93)	GND	MBDATA(94)	MBDATA(95)
22	MBDATA(96)	MBDATA(97)	MBDATA(98)	MBDATA(99)	MBDATA(100)
23	MBDATA(101)	MBDATA(102)	MBDATA(103)	MBDATA(104)	MBDATA(105)
24	MBDATA(106)	MBDATA(107)	MBDATA(108)	MBDATA(109)	MBDATA(110)
25	MBDATA(111)	MBDATA(112)	MBDATA(113)	MBDATA(114)	MBDATA(115
26	MBDATA(116)	MBDATA(117)	GND	MBDATA(118)	MBDATA(119)
27	MBDATA(120)	MBDATA(121)	MBDATA(122)	MBDATA(123)	MBDATA(124)
28	MBDATA(125)	MBDATA(126)	MBDATA(127)	MBAD(0)	MBAD(1)
29	MBAD(2)	MBAD(3)	MBAD(4)	MBAD(5)	MBAD(6)
30	MBAD(7)	MBAD(8)	MBAD(9)	MBAD(10)	MBAD(11)
31	MBAD(12)	MBAD(13)	GND	MBAD(14)	MBAD(15)
32	MBAD(16)	MBAD(17)	MBAD(18)	MBAD(19)	MBAD(20)
33	MBAD(21)	MBAD(22)	MBAD(23)	MBAD(24)	MBAD(25)
34					
35					
36					
37	MBAD(26)	MBAD(27)	MBAD(28)	MBAD(29)	MBAD(30)
38	MBAD(31)	MBRESET	RD/WR*	DSTROBE*	DDONE*
39	Reserved	BOSS	GND	BOSSREQ	BOSSGROUT
40	Reserved	AP_FIFO_EMTY	START_LOAD*	Reserved	BOSSGRIN
41	BUFFER(1)	BUFFER(0)	MOD_DONE(0)	MOD_DONE(1)	MOD_DONE(2)
42	MOD_DONE(3)	MOD_DONE(4)	MOD_DONE(5)	MOD_DONE(6)	MOD_DONE(7)
43	MOD_DONE(8)	MOD_DONE(9)	MOD_DONE(10)	MOD_DONE(11)	MOD_DONE(12)
44	MOD_DONE(13)	MOD_DONE(14)	GND	MOD_DONE(15)	MOD_DONE(16)
45	MOD_DONE(17)	MOD_DONE(18)	Reserved	DONE_OUT	Reserved
46	EV_LOADED(0)	EV_LOADED(1)	EV_LOADED(2)	EV_LOADED(3)	Reserved
47	GND	GND	GND	GND	GND
PIN	A	B	C	D	E

Original Boss logic – sensitive to noise on BOSSGRIN



Modified Boss logic – to remove noise sensitivity from BOSSGRIN
USE J-K FLIP FLOP