trigger_manager

Address width: 32

Data width: 32

Base address: 0x24220000

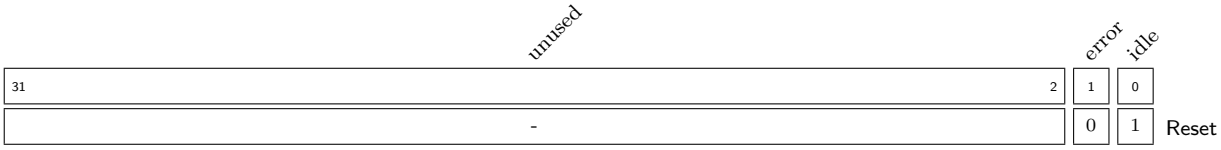
Management of global trigger and pulse signals. Also functions as a synchronizer for both ALPIDEs and RUs. Can transmit various reset to all ALPIDEs connected to RU, and reset the RU absolute time trackker. Module has 3 signal inputs that is to be connected to external FPGA IO: trigger_init (initiates trigger or pulse sequence), sync (transmits BCRST to all ALPIDEs and resets RU absolute time), spill_inc (increments SPILL ID). The module generates a 25 ns pulse from each external signal, so the external signal length is arbitrary as long as the period is over 25 ns. The module is also used for tagging pRU data, see register specifications.

1 Register List

#	Name	Mode	Address	Type	Length	Reset
0	status	RO	0x00000000	FIELDS	2	0x1
1	mode	RW	0x00000004	SLV	2	0x0
2	num_triggers	RW	0x00000008	DEFAULT	32	0x1
3	pre_cmd_delay	RW	0x0000000C	DEFAULT	32	0x0
4	trigger_delay	RW	0x00000010	DEFAULT	32	0xE
5	pulse_delay	RW	0x00000014	DEFAULT	32	0xE
6	pulse_trigger_delay	RW	0x00000018	DEFAULT	32	0xE
7	trigger_init	PULSE	0x0000001C	SL	1	0x0
8	num_trains	RW	0x00000020	DEFAULT	32	0x1
9	trains_delay	RW	0x00000024	DEFAULT	32	0xE
10	alpide_grst	PULSE	0x00000028	SL	1	0x0
11	alpide_prst	PULSE	0x0000002C	SL	1	0x0
12	alpide_bcrst	PULSE	0x00000030	SL	1	0x0
13	alpide_rorst	PULSE	0x00000034	SL	1	0x0
14	trigger_source	RW	0x00000038	SLV	2	0x3
15	alpide_mode	RW	0x0000003C	SLV	1	0x1
16	absolute_time	RO	0x00000040	DEFAULT	32	0x0
17	reset_time	PULSE	0x00000044	SL	1	0x0
18	spill_id	RO	0x00000048	SLV	16	0x0
19	increment_spill_id	PULSE	0x0000004C	SL	1	0x0
20	reset_spill_id	PULSE	0x00000050	SL	1	0x0

2 Registers

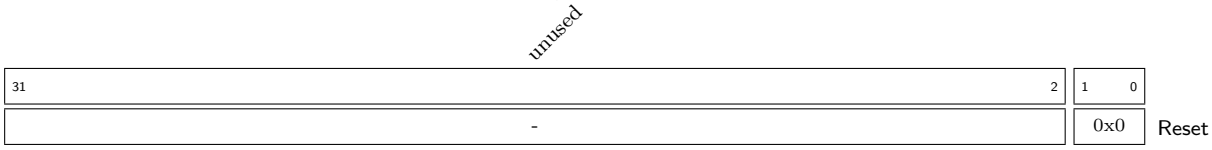
Register 2.1: STATUS - RO (0x00000000)
Status of trigger/pulse transmits.



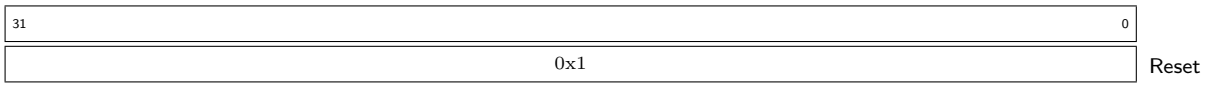
idle Indicates that the manager is free to receive new commands.

error Indicates that the register settings prevents any commands to be transmitted.

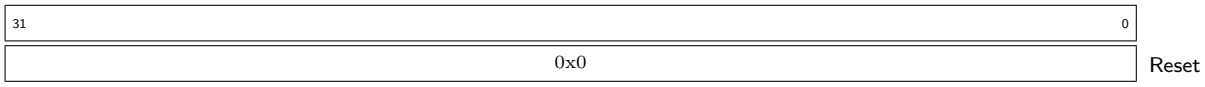
Register 2.2: MODE - RW (0x00000004)
Controls whether transmitting trigger or pulse opcode. 0 = trigger, 1 = pulse, 2 = pulse and trigger combo, 3 = no action.



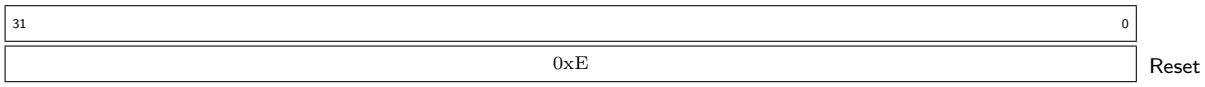
Register 2.3: NUM_TRIGGERS - RW (0x00000008)
The number of trigger/pulse/pulse-trigger-combos to be transmitted.



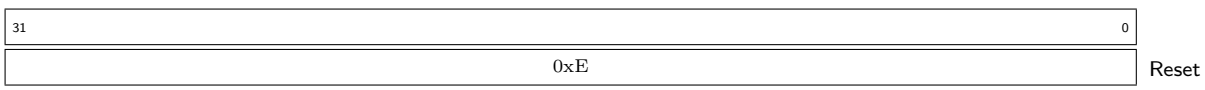
Register 2.4: PRE_CMD_DELAY - RW (0x0000000C)
Optional delay before the transmission of commands after initiation (either register or external signal). Not added to delay between command in train (when num_triggers is larger than one).



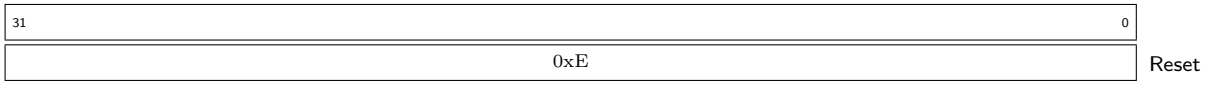
Register 2.5: TRIGGER_DELAY - RW (0x00000010)
For mode(0) - The number of clock cycles (40 MHz) between each trigger command. Minimum 14 (time required to complete operation).



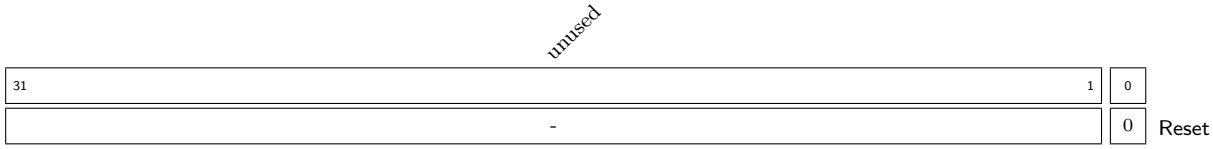
Register 2.6: PULSE_DELAY - RW (0x00000014)
For mode(1,2) - The number of clock cycles (40 MHz) between each pulse command. Minimum 14 (time required to complete operation). For mode(2,3) this must be larger than pulse_trigger_gap + 14!



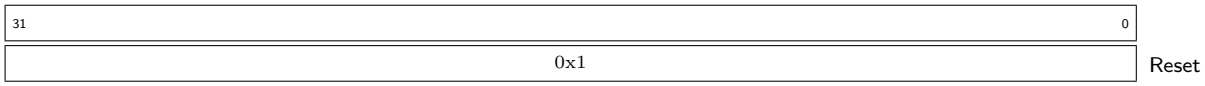
Register 2.7: PULSE_TRIGGER_DELAY - RW (0x00000018)
For mode(2) - The number of clock cycles (40 MHz) between the pulse and the trigger command. Minimum 14 (time required to complete operation).



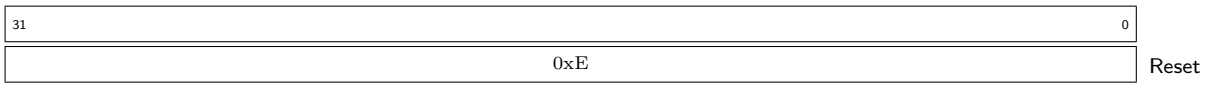
Register 2.8: TRIGGER_INIT - PULSE FOR 1 CYCLES - (0x0000001C)
Initiates the command transmission process for all modes.



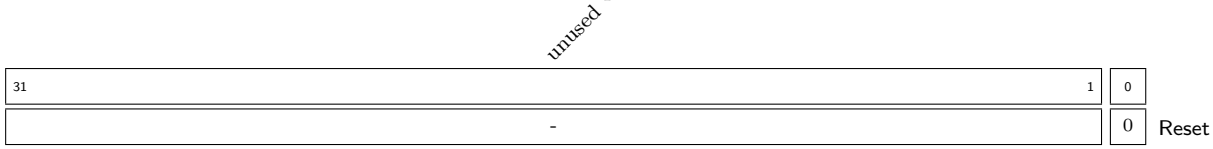
Register 2.9: NUM_TRAINS - RW (0x00000020)
The number of trains to be transmitted.



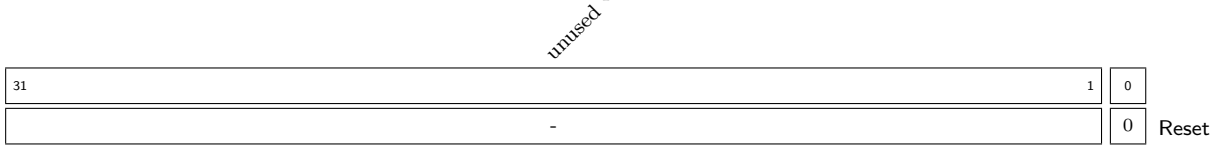
Register 2.10: TRAINS_DELAY - RW (0x00000024)
The delay between each train - from train stop to it begins again - in 40 MHz clock cycles. Minimum 14 (time required to complete operation).



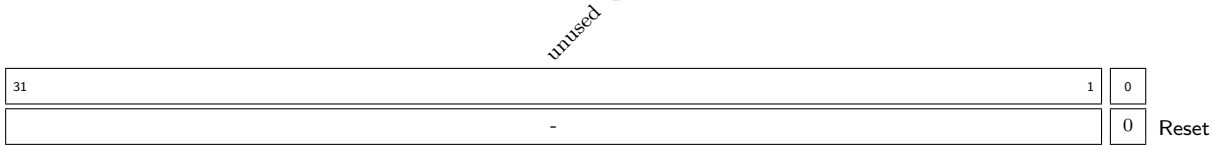
Register 2.11: ALPIDE_GRST - PULSE FOR 1 CYCLES - (0x00000028)
Transmit the GRST opcode to all ALPIDEs.



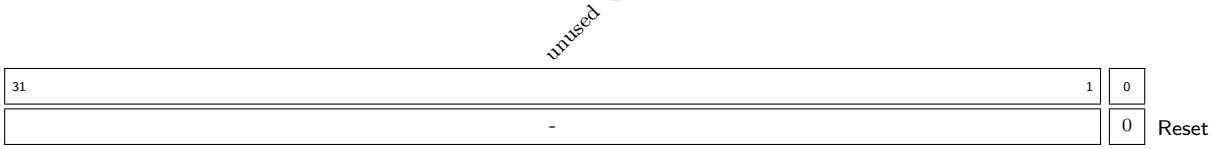
Register 2.12: ALPIDE_PRST - PULSE FOR 1 CYCLES - (0x0000002C)
Transmit the PRST opcode to all ALPIDEs.



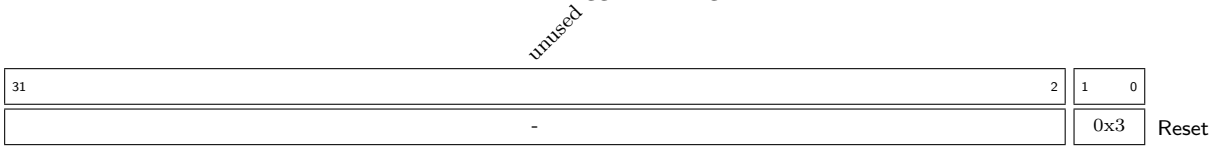
Register 2.13: ALPIDE_BCRST - PULSE FOR 1 CYCLES - (0x00000030)
Transmit the BCRST opcode to all ALPIDEs.



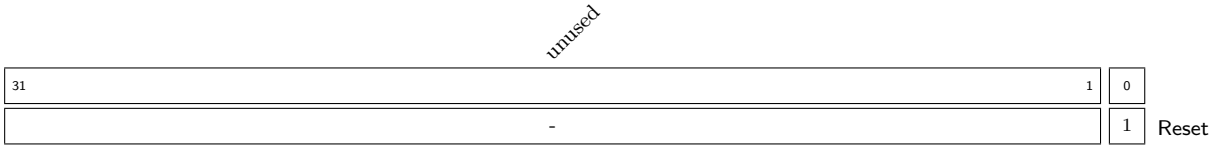
Register 2.14: ALPIDE_RORST - PULSE FOR 1 CYCLES - (0x00000034)
Transmit the RORST opcode to all ALPIDEs.



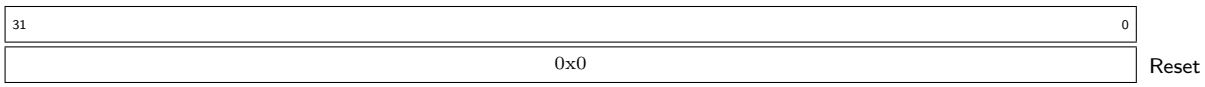
Register 2.15: TRIGGER_SOURCE - RW (0x00000038)
Optional, used for data tagging. The source of the ALPIDE trigger signal. 0x0 = ALPIDE Internal Strobe Sequencer, 0x1 = External pRU Hardware Signal, 0x2 = Software trigger. Used in data tagging, 0x3 = FPGA Trigger Manager.



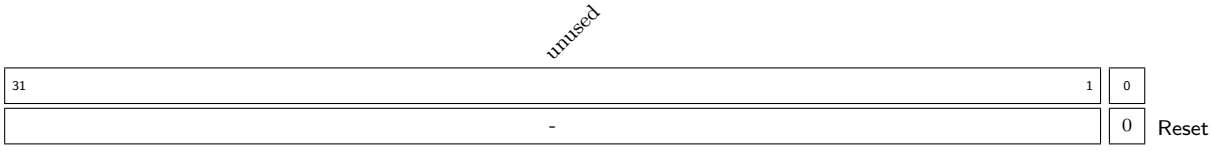
Register 2.16: ALPIDE_MODE - RW (0x0000003C)
Optional, used for data tagging. The readout mode the ALPIDEs are configured in. 0x0 = TRIGGERED mode, 0x1 = CONTINUOUS mode. Used in data tagging.



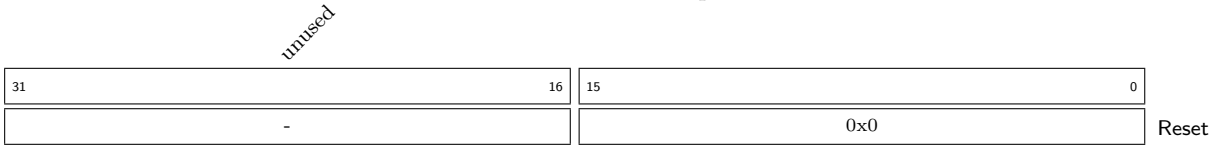
Register 2.17: ABSOLUTE_TIME - RO (0x00000040)
32-bit counter value of 120MHz system clock. Used to tag data.



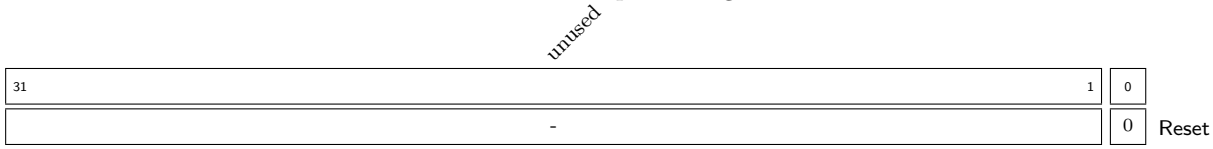
Register 2.18: RESET_TIME - PULSE FOR 1 CYCLES - (0x00000044)
Resets the absolute time register.



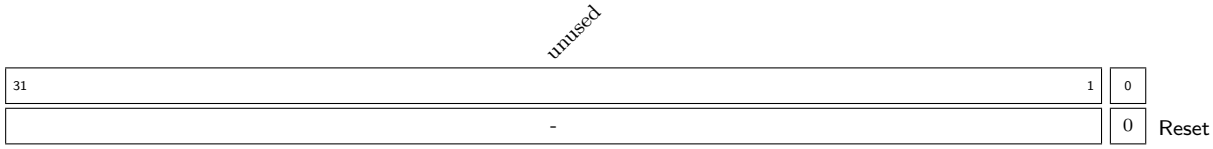
Register 2.19: SPILL_ID - RO (0x00000048)
The ID of current spill.



Register 2.20: INCREMENT_SPILL_ID - PULSE FOR 1 CYCLES - (0x0000004C)
Increments the spill ID register.



Register 2.21: RESET_SPILL_ID - PULSE FOR 1 CYCLES - (0x00000050)
Increments the spill ID register.



3 Example VHDL Register Access

All registers are bundled in records based on their mode. E.g. all RW registers are accessed through the record *bustype_rw_regs*. Access is also dependent on the type of register. All register of type SL, SLV and DEFAULT are all directly accessed by just specifying the mode record signal. E.g. the RW register *reg0* can be assigned a value like this (assuming AXI-bus):

```
axi_rw_regs.reg0 <= (others => '0');
```

Registers of type FIELD cannot be directly accessed without specification of a certain field. This is because the registers are implemented as a record in VHDL (thus a record of records). E.g. if the RO register *reg1* contains the field *field3* it can be accessed like this (assuming AXI-bus):

```
axi_ro_regs.reg1.field3 <= (others => '0');
```