Customize Pin

axi-stream (1.0)

Layered Metadata Summary

No Metadata defined

NOTE: Slave interfaces may not define
Metadata until Validation has been run

Component Name S_AXIS_S2MM

Signal Properties

Has Tlast (Auto)	true
Has Tkeep (Auto)	true
Has Tstrb (Auto)	false
Has Tready (Auto)	true
Tuser Width (Auto)	0
Tid Width (Auto)	0
Tdest Width (Auto)	0
Tdata Num Bytes (Auto)	4

Clock Properties

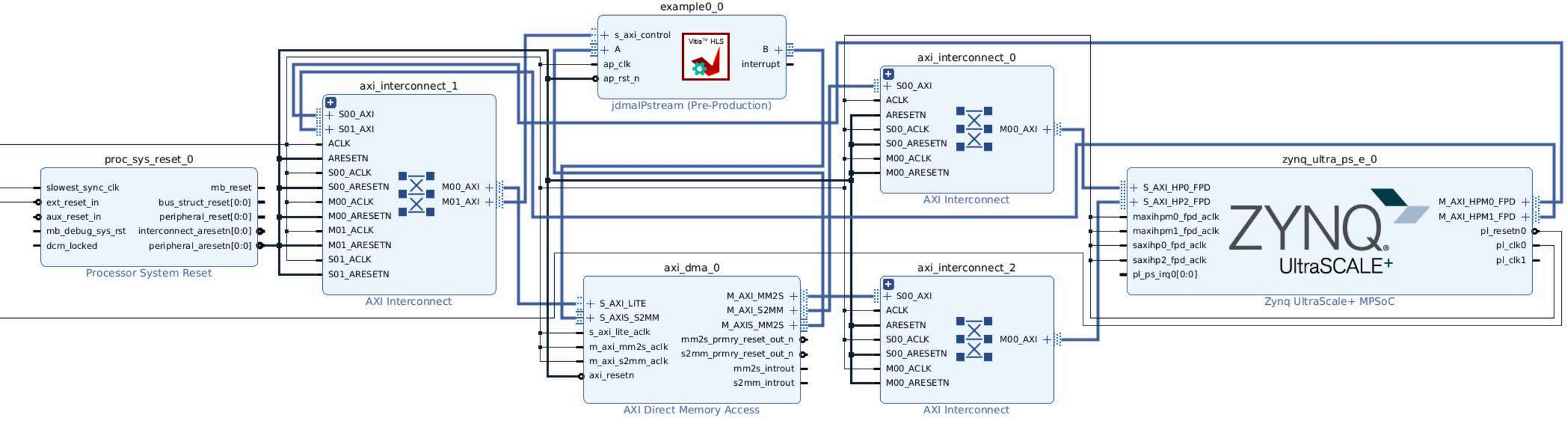
Clk Domain (Auto)	ltra_ps_e_0_1_pl_clk0
Phase (Auto)	0.0
Frequency (MHz) (Auto)	99.999001

Layered Data

Layered Metadata (Auto)	undef

OK

Cancel


example0_0
s_axi_control
A
ap_clk
ap_rst_n
B
interrupt
jdmaIPstream (Pre-Production)
axi_interconnect_1
S00_AXI
S01_AXI
ACLK
ARESETN
S00_ACLK
S00_ARESETN
M00_ACLK
M00_ARESETN
M01_ACLK
M01_ARESETN
S01_ACLK
S01_ARESETN
M00_AXI
M01_AXI
AXI Interconnect
proc_sys_reset_0
slowest_sync_clk
ext_reset_in
aux_reset_in
mb_debug_sys_rst
dcm_locked
mb_reset
bus_struct_reset[0:0]
peripheral_reset[0:0]
interconnect_aresetn[0:0]
peripheral_aresetn[0:0]
Processor System Reset
axi_interconnect_0
S00_AXI
ACLK
ARESETN
S00_ACLK
S00_ARESETN
M00_ACLK
M00_ARESETN
M00_AXI
AXI Interconnect
axi_dma_0
S_AXI_LITE
S_AXIS_S2MM
s_axi_lite_aclk
m_axi_mm2s_aclk
m_axi_s2mm_aclk
axi_resetn
M_AXI_MM2S
M_AXI_S2MM
M_AXIS_MM2S
mm2s_prmry_reset_out_n
s2mm_prmry_reset_out_n
mm2s_introut
s2mm_introut
AXI Direct Memory Access
axi_interconnect_2
S00_AXI
ACLK
ARESETN
S00_ACLK
S00_ARESETN
M00_ACLK
M00_ARESETN
M00_AXI
AXI Interconnect
zynq_ultra_ps_e_0
S_AXI_HP0_FPD
S_AXI_HP2_FPD
maxihpm0_fpd_aclk
maxihpm1_fpd_aclk
saxihp0_fpd_aclk
saxihp2_fpd_aclk
pl_ps_irq0[0:0]
M_AXI_HPM0_FPD
M_AXI_HPM1_FPD
pl_resetn0
pl_clk0
pl_clk1
Zynq UltraScale+ MPSoC

Copy the design files to the board

Rename and copy the BIT file and HWH file created by Vivado to a folder on your board. I renamed the files to:

- dma_axis_ip_example.bit
- dma_axis_ip_example.hwh

If you downloaded this notebook from GitHub you can also copy it notebook to the same folder. Alternatively, you can create a new Jupyter Notebook and copy the code into the new notebook to run it.

Instantiate and download the overlay

In [19]:
```
from pynq import Overlay

ol = Overlay("./dma_axis_ip_example.bit")
```

We can check the IPs in this overlay using the IP dictionary (*ip_dict*).

In [20]:
```
ol.ip_dict
```

Out[20]:

```
{'axi_dma_0': {'type': 'xilinx.com:ip:axi_dma:7.1',
  'mem_id': 'S_AXI_LITE',
  'memtype': 'REGISTER',
  'gpio': {},
  'interrupts': {},
  'parameters': {'C_S_AXI_LITE_ADDR_WIDTH': '10',
   'C_S_AXI_LITE_DATA_WIDTH': '32',
   'C_DLYTMR_RESOLUTION': '125',
   'C_PRMRY_IS_ACLK_ASYNC': '0',
   'C_ENABLE_MULTI_CHANNEL': '0',
   'C_NUM_MM2S_CHANNELS': '1',
   'C_NUM_S2MM_CHANNELS': '1',
   'C_INCLUDE_SG': '0',
   'C_SG_INCLUDE_STSCNTRL_STRM': '0',
   'C_SG_USE_STSAPP_LENGTH': '0',
   'C_SG_LENGTH_WIDTH': '26',
   'C_M_AXI_SG_ADDR_WIDTH': '32',
   'C_M_AXI_SG_DATA_WIDTH': '32',
   'C_M_AXIS_MM2S_CNTRL_TDATA_WIDTH': '32',
   'C_S_AXIS_S2MM_STS_TDATA_WIDTH': '32',
   'C_MICRO_DMA': '0',
   'C_INCLUDE_MM2S': '1',
   'C_INCLUDE_MM2S_SF': '1',
   'C_MM2S_BURST_SIZE': '16',
   'C_M_AXI_MM2S_ADDR_WIDTH': '32',
   'C_M_AXI_MM2S_DATA_WIDTH': '32',
   'C_M_AXIS_MM2S_TDATA_WIDTH': '32',
   'C_INCLUDE_MM2S_DRE': '0',
   'C_INCLUDE_S2MM': '1',
   'C_INCLUDE_S2MM_SF': '1',
   'C_S2MM_BURST_SIZE': '16',
   'C_M_AXI_S2MM_ADDR_WIDTH': '32',
   'C_M_AXI_S2MM_DATA_WIDTH': '32',
   'C_S_AXIS_S2MM_TDATA_WIDTH': '32',
   'C_INCLUDE_S2MM_DRE': '0',
   'C_INCREASE_THROUGHPUT': '0',
   'C_FAMILY': 'zynquplus',
   'Component_Name': 'design_1_axi_dma_0_0',
   'c_include_sg': '0',
   'c_enable_multi_channel': '0',
   'c_num_mm2s_channels': '1',
   'c_num_s2mm_channels': '1',
   'c_sg_length_width': '26',
   'c_dlytmr_resolution': '125',
   'c_prmry_is_aclk_async': '0',
   'c_sg_include_stscntrl_strm': '0',
   'c_micro_dma': '0',
   'c_include_mm2s': '1',
   'c_m_axi_mm2s_data_width': '32',
   'c_m_axis_mm2s_tdata_width': '32',
   'c_include_mm2s_dre': '0',
   'c_include_mm2s_sf': '1',
   'c_mm2s_burst_size': '16',
   'c_include_s2mm': '1',
   'c_sg_use_stsapp_length': '0',
   'c_m_axi_s2mm_data_width': '32',
   'c_s_axis_s2mm_tdata_width': '32',
   'c_include_s2mm_dre': '0',
   'c_include_s2mm_sf': '1',
   'c_s2mm_burst_size': '16',
   'c_addr_width': '32',
   'c_single_interface': '0',
   'c_increase_throughput': '0',
   'EDK_IPTYPE': 'PERIPHERAL',
   'C_BASEADDR': '0xA0000000',
   'C_HIGHADDR': '0xA000FFFF',
   'DATA_WIDTH': '32',
   'PROTOCOL': 'AXI4LITE',
   'FREQ_HZ': '99999001',
   'ID_WIDTH': '0',
   'ADDR_WIDTH': '10',
   'AWUSER_WIDTH': '0',
   'ARUSER_WIDTH': '0',
   'WUSER_WIDTH': '0',
   'RUSER_WIDTH': '0',
   'BUSER_WIDTH': '0',
   'READ_WRITE_MODE': 'READ_WRITE',
   'HAS_BURST': '0',
   'HAS_LOCK': '0',
   'HAS_PROT': '0',
   'HAS_CACHE': '0',
   'HAS_QOS': '0',
   'HAS_REGION': '0',
   'HAS_WSTRB': '0',
   'HAS_BRESP': '1',
   'HAS_RRESP': '1',
   'SUPPORTS_NARROW_BURST': '0',
   'NUM_READ_OUTSTANDING': '2',
   'NUM_WRITE_OUTSTANDING': '2',
   'MAX_BURST_LENGTH': '1',
   'PHASE': '0.0',
   'CLK_DOMAIN': 'design_1_zynq_ultra_ps_e_0_1_pl_clk0',
   'NUM_READ_THREADS': '1',
```

```
    'NUM_WRITE_THREADS': '1',
    'RUSER_BITS_PER_BYTE': '0',
    'WUSER_BITS_PER_BYTE': '0',
    'INSERT_VIP': '0',
    'TDATA_NUM_BYTES': '4',
    'TDEST_WIDTH': '0',
    'TID_WIDTH': '0',
    'TUSER_WIDTH': '0',
    'HAS_TREADY': '1',
    'HAS_TSTRB': '0',
    'HAS_TKEEP': '1',
    'HAS_TLAST': '1',
    'LAYERED_METADATA': 'undef'},
  'registers': {'MM2S_DMACR': {'address_offset': 0,
    'size': 32,
    'access': 'read-write',
    'description': 'MM2S DMA Control Register',
    'fields': {'RS': {'bit_offset': 0,
      'bit_width': 1,
      'access': 'read-write',
      'description': 'Run / Stop control for controlling running and stopping of the DMA channel.\n  0 - Stop - DMA s
tops when current (if any) DMA operations are complete. For Scatter / Gather Mode pending commands/transfers are flus
hed or completed. \n  AXI4-Stream outs are potentially terminated early. Descriptors in the update queue are allowed
to finish updating to remote memory before engine halt.\n  For Direct Register mode pending commands/transfers are fl
ushed or completed. AXI4-Stream outs are potentially terminated.\n  The halted bit in the DMA Status register asserts
to 1 when the DMA engine is halted. This bit is cleared by AXI DMA hardware when an error occurs. The CPU can also ch
oose to clear this bit to stop DMA operations.\n  1 - Run - Start DMA operations. The halted bit in the DMA Status re
gister deasserts to 0 when the DMA engine begins operations.\n'},
     'Reset': {'bit_offset': 2,
      'bit_width': 1,
      'access': 'read-write',
      'description': 'Soft reset for resetting the AXI DMA core. Setting this bit to a 1 causes the AXI DMA to be res
et. Reset is accomplished gracefully. Pending commands/transfers are flushed or completed.\nAXI4-Stream outs are pote
ntially terminated early. Setting either MM2S_DMACR. Reset = 1 or S2MM_DMACR.Reset = 1 resets the entire AXI DMA engi
ne. After completion of a soft reset, all registers and bits are in the Reset State.  0 - Normal operation.  1 -
Reset in progress.\n'},
     'Keyhole': {'bit_offset': 3,
      'bit_width': 1,
      'access': 'read-write',
      'description': 'Keyhole Read. Setting this bit to 1 causes AXI DMA to initiate MM2S reads (AXI4read) in non-inc
rementing address mode (Fixed Address Burst transfer on AXI4). This bit can be updated when AXI DMA is in idle. When
using keyhole operation the Max Burst Length should not exceed 16. This bit should not be set when DRE is enabled.\nT
his bit is non functional when the multichannel feature is enabled or in Direct Register mode.\n'},
     'Cyclic_BD_Enable': {'bit_offset': 4,
      'bit_width': 1,
      'access': 'read-write',
      'description': 'When set to 1, the DMA operates in Cyclic Buffer Descriptor (BD) mode without any user interven
tion. In this mode, the Scatter Gather module ignores the Completed bit of the BD. With this bit set, you can use the
same BDs in cyclic manner without worrying about any stale descriptor errors.\nThis bit should be set/unset only when
the DMA is idle or when not running. Updating this bit while the DMA is running can result in unexpected behavior.\nT
his bit is non functional when DMA operates in multichannel mode.\n'},
     'IOC_IrqEn': {'bit_offset': 12,
      'bit_width': 1,
      'access': 'read-write',
      'description': 'Interrupt on Complete (IOC) Interrupt Enable. When set to 1, allows DMASR.IOC_Irq to generate a
n interrupt out for descriptors with the IOC bit set.  0 - IOC Interrupt disabled  1 - IOC Interrupt enabled
\n'},
     'Dly_IrqEn': {'bit_offset': 13,
      'bit_width': 1,
      'access': 'read-write',
      'description': 'Interrupt on Delay Timer Interrupt Enable. When set to 1, allows DMASR.Dly_Irq to generate an i
nterrupt out.  0 - Delay Interrupt disabled  1 - Delay Interrupt enabled Note: This field is ignored when AXI DM
A is configured for Direct Register Mode.\n'},
     'Err_IrqEn': {'bit_offset': 14,
      'bit_width': 1,
      'access': 'read-write',
      'description': 'Interrupt on Error Interrupt Enable.\n  0 - Error Interrupt disabled\n  1 - Error Interrupt ena
bled\n'},
     'IRQThreshold': {'bit_offset': 16,
      'bit_width': 8,
      'access': 'read-write',
      'description': 'Interrupt Threshold. This value is used for setting the interrupt threshold. When IOC interrupt
events occur, an internal counter counts down from the Interrupt Threshold setting. When the count reaches zero, an i
nterrupt out is generated by the DMA engine.  Note: The minimum setting for the threshold is 0x01. A write of 0x00 t
o this register has no effect.  Note: This field is ignored when AXI DMA is configured for Direct Register Mod
e.\n'},
     'IRQDelay': {'bit_offset': 24,
      'bit_width': 8,
      'access': 'read-write',
      'description': 'Interrupt Delay Time Out. This value is used for setting the interrupt timeout value. The inter
rupt timeout mechanism causes the DMA engine to generate an interrupt after the delay time period has expired. Timer
begins counting at the end of a packet and resets with receipt of a new packet or a timeout event occurs.\nNote: Sett
ing this value to zero disables the delay timer interrupt.\nNote: This field is ignored when AXI DMA is configured fo
r Direct Register Mode.\n'}}},
   'MM2S_DMASR': {'address_offset': 4,
    'size': 32,
    'access': 'read-write',
    'description': 'MM2S DMA Status Register',
    'fields': {'Halted': {'bit_offset': 0,
      'bit_width': 1,
      'access': 'read-only',
      'description': 'DMA Channel Halted. Indicates the run/stop state of the DMA channel.  0 - DMA channel runni
ng.  1 - DMA channel halted.  For Scatter / Gather Mode this bit gets set when DMACR.RS = 0 and DMA and SG operat
```

```
ions have halted. For Direct Register mode (C_INCLUDE_SG = 0) this bit gets set when DMACR.RS = 0 and DMA operations have halted. There can be a lag of time between when DMACR.RS = 0 and when DMASR.Halted = 1 Note: When halted (RS= 0 and Halted = 1), writing to CURDESC_PTR or TAILDESC_PTR pointer registers has no effect on DMA operations when in Sca tter Gather Mode. For Direct Register Mode, writing to the LENGTH register has no effect on DMA operations.\n'},
     'Idle': {'bit_offset': 1,
      'bit_width': 1,
      'access': 'read-only',
      'description': 'DMA Channel Idle. Indicates the state of AXI DMA operations.\nFor Scatter / Gather Mode when ID LE indicates the SG Engine has reached the tail pointer for the associated channel and all queued descriptors have be en processed. Writing to the tail pointer register automatically restarts DMA operations.\nFor Direct Register Mode w hen IDLE indicates the current transfer has completed.      0 - Not Idle. For Scatter / Gather Mode, SG has not reach ed tail descriptor pointer and/or DMA operations in progress. For Direct Register Mode, transfer is not complete.
1 - Idle. For Scatter / Gather Mode, SG has reached tail descriptor pointer and DMA operation paused. for Direct Regi ster Mode, DMA transfer has completed and controller is paused.  Note: This bit is 0 when channel is halted (DMASR.Ha lted=1). This bit is also 0 prior to initial transfer when AXI DMA configured for Direct Register Mode.\n'},
     'SGIncld': {'bit_offset': 3,
      'bit_width': 1,
      'access': 'read-only',
      'description': '1 - Scatter Gather Enabled\n0 - Scatter Gather not enabled\n'},
     'DMAIntErr': {'bit_offset': 4,
      'bit_width': 1,
      'access': 'read-only',
      'description': 'DMA Internal Error. Internal error occurs if the buffer length specified in the fetched descrip tor is set to 0. This error condition causes the AXI DMA to halt gracefully. The DMACR.RS bit is set to 0, and when t he engine has completely shut down, the DMASR.Halted bit is set to 1.      0 - No DMA Internal Errors       1 - DMA Internal Error detected. DMA Engine halts\n'},
     'DMASlvErr': {'bit_offset': 5,
      'bit_width': 1,
      'access': 'read-only',
      'description': 'DMA Slave Error. This error occurs if the slave read from the Memory Map interface issues a Sla ve Error. This error condition causes the AXI DMA to halt gracefully. The DMACR.RS bit is set to 0, and when the engi ne has completely shut down, the DMASR.Halted bit is set to 1.      0 - No DMA Slave Errors.      1 - DMA Slave Error detected. DMA Engine halts\n'},
     'DMADecErr': {'bit_offset': 6,
      'bit_width': 1,
      'access': 'read-only',
      'description': 'DMA Decode Error. This error occurs if the address request points to an invalid address. This e rror condition causes the AXI DMA to halt gracefully. The DMACR.RS bit is set to 0, and when the engine has completel y shut down, the DMASR.Halted bit is set to 1.      0 - No DMA Decode Errors.      1 - DMA Decode Error detected. DMA Engine halts.\n'},
     'SGIntErr': {'bit_offset': 8,
      'bit_width': 1,
      'access': 'read-only',
      'description': 'Scatter Gather Internal Error. This error occurs if a descriptor with the “Complete bit” alread y set is fetched. Refer to the Scatter Gather Descriptor section for more information.This indicates to the SG Engine that the descriptor is a stale descriptor. This error condition causes the AXI DMA to halt gracefully. The DMACR.RS b it is set to 0, and when the engine has completely shut down, the DMASR.Halted bit is set to 1.      0 - No SG Intern al Errors.      1 - SG Internal Error detected. DMA Engine halts.  Note: This bit is not used and is fixed at 0 when AXI DMA is configured for Direct Register Mode.\n'},
     'SGSlvErr': {'bit_offset': 9,
      'bit_width': 1,
      'access': 'read-only',
      'description': 'Scatter Gather Slave Error. This error occurs if the slave read from on the Memory Map interfac e issues a Slave error. This error condition causes the AXI DMA to halt gracefully. The DMACR.RS bit is set to 0, and when the engine has completely shut down, the DMASR.Halted bit is set to 1.      0 - No SG Slave Errors.      1 - SG Slave Error detected. DMA Engine halts.  Note: This bit is not used and is fixed at 0 when AXI DMA is configured for Direct Register Mode. \n'},
     'SGDecErr': {'bit_offset': 10,
      'bit_width': 1,
      'access': 'read-only',
      'description': 'Scatter Gather Decode Error. This error occurs if CURDESC_PTR and/or NXTDESC_PTR points to an i nvalid address. This error condition causes the AXI DMA to halt gracefully. The DMACR.RS bit is set to 0, and when th e engine has completely shut down, the DMASR.Halted bit is set to 1.      0 - No SG Decode Errors.      1 - SG Decode Error detected. DMA Engine halts.  Note: This bit is not used and is fixed at 0 when AXI DMA is configured for Direct Register Mode. \n'},
     'IOC_Irq': {'bit_offset': 12,
      'bit_width': 1,
      'access': 'read-write',
      'description': 'Interrupt on Complete. When set to 1 for Scatter/Gather Mode, indicates an interrupt event was generated on completion of a descriptor. This occurs for descriptors with the End of Frame (EOF) bit set. When set to 1 for Direct Register Mode, indicates an interrupt event was generated on completion of a transfer. If the correspond ing bit is enabled in the MM2S_DMACR (IOC_IrqEn = 1) and if the interrupt threshold has been met, causes an interrupt out to be generated from the AXI DMA.      0 - No IOC Interrupt.      1 - IOC Interrupt detected.  Writing a 1 to thi s bit will clear it.\n'},
     'Dly_Irq': {'bit_offset': 13,
      'bit_width': 1,
      'access': 'read-write',
      'description': 'Interrupt on Delay. When set to 1, indicates an interrupt event was generated on delay timer ti me out. If the corresponding bit is enabled in the MM2S_DMACR (Dly_IrqEn = 1), an interrupt out is generated from the AXI DMA.      0 - No Delay Interrupt.      1 - Delay Interrupt detected. Note: This bit is not used and is fixed at 0 when AXI DMA is configured for Direct Register Mode. \n'},
     'Err_Irq': {'bit_offset': 14,
      'bit_width': 1,
      'access': 'read-write',
      'description': 'Interrupt on Error. When set to 1, indicates an interrupt event was generated on error. If the corresponding bit is enabled in the MM2S_DMACR (Err_IrqEn = 1), an interrupt out is generated from the AXI DMA.\nWrit ing a 1 to this bit will clear it.   \n0 - No error Interrupt.   \n1 - Error interrupt detected.\n'},
     'IRQThresholdSts': {'bit_offset': 16,
      'bit_width': 8,
      'access': 'read-only',
      'description': 'Interrupt Threshold Status. Indicates current interrupt threshold value.\nNote: Applicable only when Scatter Gather is enabled.\n'},
     'IRQDelaySts': {'bit_offset': 24,
```

```
      'bit_width': 8,
      'access': 'read-only',
      'description': 'Interrupt Delay Time Status. Indicates current interrupt delay time value.\nNote: Applicable on
ly when Scatter Gather is enabled.\n'}}},
   'MM2S_CURDESC': {'address_offset': 8,
    'size': 32,
    'access': 'read-write',
    'description': 'MM2S DMA Current Descriptor Pointer Register',
    'fields': {'Current_Descriptor_Pointer': {'bit_offset': 6,
      'bit_width': 26,
      'access': 'read-write',
      'description': 'Indicates the pointer of the current descriptor being worked on. This register must contain a p
ointer to a valid descriptor prior to writing the TAILDESC_PTR register. Otherwise, undefined results occur. When DMA
CR.RS is 1, CURDESC_PTR becomes Read Only (RO) and is used to fetch the first descriptor.\nWhen the DMA Engine is run
ning (DMACR.RS=1), CURDESC_PTR registers are updated by AXI DMA to indicate the current descriptor being worked on.\n
On error detection, CURDESC_PTR is updated to reflect the descriptor associated with the detected error.\nNote: The r
egister can only be written to by the CPU when the DMA Engine is Halted (DMACR.RS=0 and DMASR.Halted =1). At all othe
r times, this register is Read Only (RO). Descriptors must be 16 word aligned, that is, 0x00, 0x40, 0x80 and others.
Any other alignment has undefined results.\n'}}},
   'MM2S_CURDESC_MSB': {'address_offset': 12,
    'size': 32,
    'access': 'read-write',
    'description': 'MM2S DMA Current Descriptor Pointer Register',
    'fields': {'Current_Descriptor_Pointer': {'bit_offset': 0,
      'bit_width': 32,
      'access': 'read-write',
      'description': 'Indicates the pointer of the current descriptor being worked on. This register must contain a p
ointer to a valid descriptor prior to writing the TAILDESC_PTR register. Otherwise, undefined results occur. When DMA
CR.RS is 1, CURDESC_PTR becomes Read Only (RO) and is used to fetch the first descriptor.\nWhen the DMA Engine is run
ning (DMACR.RS=1), CURDESC_PTR registers are updated by AXI DMA to indicate the current descriptor being worked on.\n
On error detection, CURDESC_PTR is updated to reflect the descriptor associated with the detected error.\nNote: The r
egister can only be written to by the CPU when the DMA Engine is Halted (DMACR.RS=0 and DMASR.Halted =1). At all othe
r times, this register is Read Only (RO). Descriptors must be 16 word aligned, that is, 0x00, 0x40, 0x80 and others.
Any other alignment has undefined results.\n'}}},
   'MM2S_TAILDESC': {'address_offset': 16,
    'size': 32,
    'access': 'read-write',
    'description': 'MM2S DMA Tail Descriptor Pointer Register',
    'fields': {'Tail_Descriptor_Pointer': {'bit_offset': 6,
      'bit_width': 26,
      'access': 'read-write',
      'description': 'Indicates the pause pointer in a descriptor chain. The AXI DMA SG Engine pauses descriptor fetc
hing after completing operations on the descriptor whose current descriptor pointer matches the tail descriptor point
er.\nWhen AXI DMA Channel is not halted (DMASR.Halted = 0), a write by the CPU to the TAILDESC_PTR register causes th
e AXI DMA SG Engine to start fetching descriptors or restart if it was idle (DMASR.Idle = 1). If it was not idle, wri
ting TAILDESC_PTR has no effect except to reposition the pause point.\nIf the AXI DMA Channel is halted (DMASR.Halted
= 1 and DMACR.RS = 0), a write by the CPU to the TAILDESC_PTR register has no effect except to reposition the pause p
oint.\nNote: The software must not move the tail pointer to a location that has not been updated. The software proces
ses and reallocates all completed descriptors (Cmplted = 1), clears the completed bits and then moves the tail pointe
r. The software must move the pointer to the last descriptor it updated. Descriptors must be 16-word aligned, that i
s, 0x00, 0x40, 0x80, and so forth. Any other alignment has undefined results. \n'}}},
   'MM2S_TAILDESC_MSB': {'address_offset': 20,
    'size': 32,
    'access': 'read-write',
    'description': 'MM2S DMA Tail Descriptor Pointer Register',
    'fields': {'Tail_Descriptor_Pointer': {'bit_offset': 0,
      'bit_width': 32,
      'access': 'read-write',
      'description': 'Indicates the pause pointer in a descriptor chain. The AXI DMA SG Engine pauses descriptor fetc
hing after completing operations on the descriptor whose current descriptor pointer matches the tail descriptor point
er.\nWhen AXI DMA Channel is not halted (DMASR.Halted = 0), a write by the CPU to the TAILDESC_PTR register causes th
e AXI DMA SG Engine to start fetching descriptors or restart if it was idle (DMASR.Idle = 1). If it was not idle, wri
ting TAILDESC_PTR has no effect except to reposition the pause point.\nIf the AXI DMA Channel is halted (DMASR.Halted
= 1 and DMACR.RS = 0), a write by the CPU to the TAILDESC_PTR register has no effect except to reposition the pause p
oint.\nNote: The software must not move the tail pointer to a location that has not been updated. The software proces
ses and reallocates all completed descriptors (Cmplted = 1), clears the completed bits and then moves the tail pointe
r. The software must move the pointer to the last descriptor it updated. Descriptors must be 16-word aligned, that i
s, 0x00, 0x40, 0x80, and so forth. Any other alignment has undefined results. \n'}}},
   'MM2S_SA': {'address_offset': 24,
    'size': 32,
    'access': 'read-write',
    'description': 'MM2S Source Address Register',
    'fields': {'Source_Address': {'bit_offset': 0,
      'bit_width': 32,
      'access': 'read-write',
      'description': 'Indicates the source address AXI DMA reads from to transfer data to AXI4-Stream on the MM2S Cha
nnel.\nNote: If Data Realignment Engine is included, the Source Address can be at any byte offset. If Data Realignmen
t Engine is not included, the Source Address must be MM2S Memory Map data width aligned.\n'}}},
   'MM2S_SA_MSB': {'address_offset': 28,
    'size': 32,
    'access': 'read-write',
    'description': 'MM2S Source Address Register',
    'fields': {'Source_Address': {'bit_offset': 0,
      'bit_width': 32,
      'access': 'read-write',
      'description': 'Indicates the MSB 32 bits of the source address AXI DMA reads from to transfer data to AXI4-Str
eam on the MM2S Channel.\nNote: If Data Realignment Engine is included, the Source Address can be at any byte offset.
If Data Realignment Engine is not included, the Source Address must be MM2S Memory Map data width aligned.\n'}}},
   'MM2S_LENGTH': {'address_offset': 40,
    'size': 32,
    'access': 'read-write',
    'description': 'MM2S DMA Transfer Length Register',
    'fields': {'Length': {'bit_offset': 0,
```

```
      'bit_width': 26,
      'access': 'read-write',
      'description': 'Indicates the number of bytes to transfer for the MM2S channel. Writing a non-zero value to thi
s register starts the MM2S transfer.\n'}},
   'SG_CTL': {'address_offset': 44,
    'size': 32,
    'access': 'read-write',
    'description': 'Scatter/Gather User and Cache Control Register',
    'fields': {'SG_CACHE': {'bit_offset': 0,
      'bit_width': 4,
      'access': 'read-write',
      'description': 'Scatter/Gather Cache Control. Values written in this register reflect on the m_axi_sg_arcache a
nd m_axi_sg_awcache signals of the M_AXI_SG interface.\n'},
     'SG_USER': {'bit_offset': 8,
      'bit_width': 4,
      'access': 'read-write',
      'description': 'Scatter/Gather User Control. Values written in this register reflect on the m_axi_sg_aruser and
m_axi_sg_awuser signals of the M_AXI_SG interface.\n'}}},
   'S2MM_DMACR': {'address_offset': 48,
    'size': 32,
    'access': 'read-write',
    'description': 'S2MM DMA Control Register',
    'fields': {'RS': {'bit_offset': 0,
      'bit_width': 1,
      'access': 'read-write',
      'description': 'Run / Stop control for controlling running and stopping of the DMA channel.\n  0 - Stop - DMA s
tops when current (if any) DMA operations are complete. For Scatter / Gather Mode pending commands/transfers are flus
hed or completed. \n  AXI4-Stream outs are potentially terminated early. Descriptors in the update queue are allowed
to finish updating to remote memory before engine halt.\n  For Direct Register mode pending commands/transfers are fl
ushed or completed. AXI4-Stream outs are potentially terminated. Data integrity on S2MM AXI4 cannot be guaranteed.\n
The halted bit in the DMA Status register asserts to 1 when the DMA engine is halted. This bit is cleared by AXI DMA
hardware when an error occurs. The CPU can also choose to clear this bit to stop DMA operations.\n  1 - Run - Start D
MA operations. The halted bit in the DMA Status register deasserts to 0 when the DMA engine begins operations.\n'},
     'Reset': {'bit_offset': 2,
      'bit_width': 1,
      'access': 'read-write',
      'description': 'Soft reset for resetting the AXI DMA core. Setting this bit to a 1 causes the AXI DMA to be res
et. Reset is accomplished gracefully. Pending commands/transfers are flushed or completed.\nAXI4-Stream outs are term
inated early, if necessary with associated TLAST. Setting either MM2S_DMACR.Reset = 1 or S2MM_DMACR.Reset = 1 resets
the entire AXI DMA engine. After completion of a soft reset, all registers and bits are in the Reset State.      0 -
Reset not in progress. Normal operation.      1 - Reset in progress\n'},
     'Keyhole': {'bit_offset': 3,
      'bit_width': 1,
      'access': 'read-write',
      'description': 'Keyhole Write. Setting this bit to 1 causes AXI DMA to initiate S2MM writes (AXI4 Writes) in no
n-incrementing address mode (Fixed Address Burst transfer on AXI4). This bit can be modified when AXI DMA is in idle.
When enabling Key hole operation the maximum burst length cannot be more than 16. This bit should not be set when DRE
is enabled.\nThis bit is non functional when DMA is used in multichannel mode.\n'},
     'Cyclic_BD_Enable': {'bit_offset': 4,
      'bit_width': 1,
      'access': 'read-write',
      'description': 'When set to 1, the DMA operates in Cyclic Buffer Descriptor (BD) mode without any user interven
tion. In this mode, the Scatter Gather module ignores the Completed bit of the BD. With this bit set, you can use the
same BDs in cyclic manner without worrying about any stale descriptor errors.\nThis bit is non functional when DMA op
erates in Multichannel mode. or in Direct Register Mode\n'},
     'IOC_IrqEn': {'bit_offset': 12,
      'bit_width': 1,
      'access': 'read-write',
      'description': 'Interrupt on Complete (IOC) Interrupt Enable. When set to 1, allows Interrupt On Complete event
s to generate an interrupt out for descriptors with the Complete bit set.      0 - IOC Interrupt disabled      1 - IO
C Interrupt enabled\n'},
     'Dly_IrqEn': {'bit_offset': 13,
      'bit_width': 1,
      'access': 'read-write',
      'description': 'Interrupt on Delay Timer Interrupt Enable. When set to 1, allows error events to generate an in
terrupt out.      0 - Delay Interrupt disabled      1 - Delay Interrupt enabled  Note: Applicable only when Scatter G
ather is enabled.\n'},
     'Err_IrqEn': {'bit_offset': 14,
      'bit_width': 1,
      'access': 'read-write',
      'description': 'Interrupt on Error Interrupt Enable. When set to 1, allows error events to generate an interrup
t out.      0 - Error Interrupt disabled      1 - Error Interrupt enabled\n'},
     'IRQThreshold': {'bit_offset': 16,
      'bit_width': 8,
      'access': 'read-write',
      'description': 'Interrupt Threshold. This value is used for setting the interrupt threshold. When IOC interrupt
events occur, an internal counter counts down from the Interrupt Threshold setting. When the count reaches zero, an i
nterrupt out is generated by the DMA engine.\nNote: The minimum setting for the threshold is 0x01. A write of 0x00 to
this register has no effect.\nNote: Applicable only when Scatter Gather is enabled.\n'},
     'IRQDelay': {'bit_offset': 24,
      'bit_width': 8,
      'access': 'read-write',
      'description': 'Interrupt Delay Time Out. This value is used for setting the interrupt timeout value. The inter
rupt timeout mechanism causes the DMA engine to generate an interrupt after the delay time period has expired. Timer
begins counting at the end of a packet and resets with receipt of a new packet or a timeout event occurs.\nNote: Sett
ing this value to zero disables the delay timer interrupt.\nNote: Applicable only when Scatter Gather is enable
d.\n'}}},
   'S2MM_DMASR': {'address_offset': 52,
    'size': 32,
    'access': 'read-write',
    'description': 'S2MM DMA Status Register',
    'fields': {'Halted': {'bit_offset': 0,
      'bit_width': 1,
```

```
      'access': 'read-only',
      'description': 'DMA Channel Halted. Indicates the run/stop state of the DMA channel.      0 - DMA channel runni
ng.      1 - DMA channel halted.  For Scatter/Gather Mode this bit gets set when DMACR.RS = 0 and DMA and SG operatio
ns have halted. For Direct Register Mode this bit gets set when DMACR.RS = 0 and DMA operations have halted. There ca
n be a lag of time between when DMACR.RS = 0 and when DMASR.Halted = 1 \nNote: When halted (RS= 0 and Halted = 1), wr
iting to CURDESC_PTR or TAILDESC_PTR pointer registers has no effect on DMA operations when in Scatter Gather Mode. F
or Direct Register Mode, writing to the LENGTH register has no effect on DMA operations.\n'},
     'Idle': {'bit_offset': 1,
      'bit_width': 1,
      'access': 'read-only',
      'description': 'DMA Channel Idle. Indicates the state of AXI DMA operations.\nFor Scatter / Gather Mode when ID
LE indicates the SG Engine has reached the tail pointer for the associated channel and all queued descriptors have be
en processed. Writing to the tail pointer register automatically restarts DMA operations.\nFor Direct Register Mode w
hen IDLE indicates the current transfer has completed.      0 - Not Idle.      1 - Idle.   Note: This bit is 0 when c
hannel is halted (DMASR.Halted=1). This bit is also 0 prior to initial transfer when AXI DMA configured for Direct Re
gister Mode.\n'},
     'SGIncld': {'bit_offset': 3,
      'bit_width': 1,
      'access': 'read-only',
      'description': 'Scatter Gather Engine Included. DMASR.SGIncld = 1 indicates the Scatter Gather engine is includ
ed and the AXI DMA is configured for Scatter Gather mode. DMASR.SGIncld = 0 indicates the Scatter Gather engine is ex
cluded and the AXI DMA is configured for Direct Register Mode.\n'},
     'DMAIntErr': {'bit_offset': 4,
      'bit_width': 1,
      'access': 'read-only',
      'description': 'DMA Internal Error. This error occurs if the buffer length specified in the fetched descriptor
is set to 0. Also, when in Scatter Gather Mode and using the status app length field, this error occurs when the Stat
us AXI4-Stream packet RxLength field does not match the S2MM packet being received by the S_AXIS_S2MM interface. When
Scatter Gather is disabled, this error is flagged if any error occurs during Memory write or if the incoming packet i
s bigger than what is specified in the DMA length register.\nThis error condition causes the AXI DMA to halt graceful
ly. The DMACR.RS bit is set to 0, and when the engine has completely shut down, the DMASR.Halted bit is set to 1.
0 - No DMA Internal Errors      1 - DMA Internal Error detected.\n'},
     'DMASlvErr': {'bit_offset': 5,
      'bit_width': 1,
      'access': 'read-only',
      'description': 'DMA Slave Error. This error occurs if the slave read from the Memory Map interface issues a Sla
ve Error. This error condition causes the AXI DMA to halt gracefully. The DMACR.RS bit is set to 0, and when the engi
ne has completely shut down, the DMASR.Halted bit is set to 1.      0 - No DMA Slave Errors.      1 - DMA Slave Error
detected.\n'},
     'DMADecErr': {'bit_offset': 6,
      'bit_width': 1,
      'access': 'read-only',
      'description': 'DMA Decode Error. This error occurs if the address request points to an invalid address. This e
rror condition causes the AXI DMA to halt gracefully. The DMACR.RS bit is set to 0, and when the engine has completel
y shut down, the DMASR.Halted bit is set to 1.     0 - No DMA Decode Errors.   1 - DMA Decode Error detected.\n'},
     'SGIntErr': {'bit_offset': 8,
      'bit_width': 1,
      'access': 'read-only',
      'description': 'Scatter Gather Internal Error. This error occurs if a descriptor with the “Complete bit” alread
y set is fetched. This indicates to the SG Engine that the descriptor is a tail descriptor. This error condition caus
es the AXI DMA to halt gracefully. The DMACR.RS bit is set to 0, and when the engine has completely shut down, the DM
ASR.Halted bit is set to 1.      0 - No SG Internal Errors.      1 - SG Internal Error detected.  Note: Applicable on
ly when Scatter Gather is enabled. \n'},
     'SGSlvErr': {'bit_offset': 9,
      'bit_width': 1,
      'access': 'read-only',
      'description': 'Scatter Gather Slave Error. This error occurs if the slave read from on the Memory Map interfac
e issues a Slave error. This error condition causes the AXI DMA to halt gracefully. The DMACR.RS bit is set to 0, and
when the engine has completely shut down, the DMASR.Halted bit is set to 1.      0 - No SG Slave Errors.      1 - SG
Slave Error detected. DMA Engine halts. Note: Applicable only when Scatter Gather is enabled. \n'},
     'SGDecErr': {'bit_offset': 10,
      'bit_width': 1,
      'access': 'read-only',
      'description': 'Scatter Gather Decode Error. This error occurs if CURDESC_PTR and/or NXTDESC_PTR points to an i
nvalid address. This error condition causes the AXI DMA to halt gracefully. The DMACR.RS bit is set to 0, and when th
e engine has completely shut down, the DMASR.Halted bit is set to 1.      0 - No SG Decode Errors.      1 - SG Decode
Error detected. DMA Engine halts. Note: Applicable only when Scatter Gather is enabled. \n'},
     'IOC_Irq': {'bit_offset': 12,
      'bit_width': 1,
      'access': 'read-write',
      'description': 'Interrupt on Complete. When set to 1 for Scatter/Gather Mode, indicates an interrupt event was
generated on completion of a descriptor. This occurs for descriptors with the End of Frame (EOF) bit set. When set to
1 for Direct Register Mode, indicates an interrupt event was generated on completion of a transfer. If the correspond
ing bit in S2MM_DMACR is enabled (IOC_IrqEn = 1) and if the interrupt threshold has been met, causes an interrupt out
to be generated from the AXI DMA.      0 - No IOC Interrupt.      1 - IOC Interrupt detected. Writing a 1 to this bit
will clear it.\n'},
     'Dly_Irq': {'bit_offset': 13,
      'bit_width': 1,
      'access': 'read-write',
      'description': 'Interrupt on Delay. When set to 1, indicates an interrupt event was generated on delay timer ti
me out. If the corresponding bit is enabled in the S2MM_DMACR (Dly_IrqEn = 1), an interrupt out is generated from the
AXI DMA.      0 - No Delay Interrupt.      1 - Delay Interrupt detected.1 = IOC Interrupt detected. Writing a 1 to th
is bit will clear it. Note: Applicable only when Scatter Gather is enabled. \n'},
     'Err_Irq': {'bit_offset': 14,
      'bit_width': 1,
      'access': 'read-write',
      'description': 'Interrupt on Error. When set to 1, indicates an interrupt event was generated on error. If the
corresponding bit is enabled in the S2MM_DMACR (Err_IrqEn = 1), an interrupt out is generated from the AXI DMA.\nWrit
ing a 1 to this bit will clear it.      0 - No error Interrupt.      1 - Error interrupt detected.\n'},
     'IRQThresholdSts': {'bit_offset': 16,
      'bit_width': 8,
      'access': 'read-only',
      'description': 'Interrupt Threshold Status. Indicates current interrupt threshold value.\nNote: Applicable only
```

```
when Scatter Gather is enabled.\n'},
     'IRQDelaySts': {'bit_offset': 24,
      'bit_width': 8,
      'access': 'read-only',
      'description': 'Interrupt Delay Time Status. Indicates current interrupt delay time value.\nNote: Applicable on
ly when Scatter Gather is enabled.\n'}}},
   'S2MM_CURDESC': {'address_offset': 56,
    'size': 32,
    'access': 'read-write',
    'description': 'S2MM DMA Current Descriptor Pointer Register',
    'fields': {'Current_Descriptor_Pointer': {'bit_offset': 6,
      'bit_width': 26,
      'access': 'read-write',
      'description': 'Indicates the pointer of the current descriptor being worked on. This register must contain a p
ointer to a valid descriptor prior to writing the TAILDESC_PTR register. Otherwise, undefined results occur. When DMA
CR.RS is 1, CURDESC_PTR becomes Read Only (RO) and is used to fetch the first descriptor.\nWhen the DMA Engine is run
ning (DMACR.RS=1), CURDESC_PTR registers are updated by AXI DMA to indicate the current descriptor being worked on.\n
On error detection, CURDESC_PTR is updated to reflect the descriptor associated with the detected error.\nNote: The r
egister can only be written to by the CPU when the DMA Engine is Halted (DMACR.RS=0 and DMASR.Halted =1). At all othe
r times, this register is Read Only (RO). \nBuffer Descriptors must be 16 word aligned, that is, 0x00, 0x40, 0x80 and
so forth. Any other alignment has undefined results.\n'}}},
   'S2MM_CURDESC_MSB': {'address_offset': 60,
    'size': 32,
    'access': 'read-write',
    'description': 'S2MM DMA Current Descriptor Pointer Register',
    'fields': {'Current_Descriptor_Pointer': {'bit_offset': 0,
      'bit_width': 32,
      'access': 'read-write',
      'description': 'Indicates the pointer of the current descriptor being worked on. This register must contain a p
ointer to a valid descriptor prior to writing the TAILDESC_PTR register. Otherwise, undefined results occur. When DMA
CR.RS is 1, CURDESC_PTR becomes Read Only (RO) and is used to fetch the first descriptor.\nWhen the DMA Engine is run
ning (DMACR.RS=1), CURDESC_PTR registers are updated by AXI DMA to indicate the current descriptor being worked on.\n
On error detection, CURDESC_PTR is updated to reflect the descriptor associated with the detected error.\nNote: The r
egister can only be written to by the CPU when the DMA Engine is Halted (DMACR.RS=0 and DMASR.Halted =1). At all othe
r times, this register is Read Only (RO). Descriptors must be 16 word aligned, that is, 0x00, 0x40, 0x80 and others.
Any other alignment has undefined results.\n'}}},
   'S2MM_TAILDESC': {'address_offset': 64,
    'size': 32,
    'access': 'read-write',
    'description': 'S2MM DMA Tail Descriptor Pointer Register',
    'fields': {'Tail_Descriptor_Pointer': {'bit_offset': 6,
      'bit_width': 26,
      'access': 'read-write',
      'description': 'Indicates the pause pointer in a descriptor chain. The AXI DMA SG Engine pauses descriptor fetc
hing after completing operations on the descriptor whose current descriptor pointer matches the tail descriptor point
er.\nWhen AXI DMA Channel is not halted (DMASR.Halted = 0), a write by the CPU to the TAILDESC_PTR register causes th
e AXI DMA SG Engine to start fetching descriptors or restart if it was idle (DMASR.Idle = 1). If it was not idle, wri
ting TAILDESC_PTR has no effect except to reposition the pause point.\nIf the AXI DMA Channel DMACR.RS bit is set to
0 (DMASR.Halted = 1 and DMACR.RS = 0), a write by the CPU to the TAILDESC_PTR register has no effect except to reposi
tion the pause point.\nNote: The software must not move the tail pointer to a location that has not been updated. The
software processes and reallocates all completed descriptors (Cmplted = 1), clears the completed bits and then moves
the tail pointer. The software must move the pointer to the last descriptor it updated. \nDescriptors must be 16-word
aligned, that is, 0x00, 0x40, 0x80, and so forth. Any other alignment has undefined results. \n'}}},
   'S2MM_TAILDESC_MSB': {'address_offset': 68,
    'size': 32,
    'access': 'read-write',
    'description': 'S2MM DMA Tail Descriptor Pointer Register',
    'fields': {'Tail_Descriptor_Pointer': {'bit_offset': 0,
      'bit_width': 32,
      'access': 'read-write',
      'description': 'Indicates the pause pointer in a descriptor chain. The AXI DMA SG Engine pauses descriptor fetc
hing after completing operations on the descriptor whose current descriptor pointer matches the tail descriptor point
er.\nWhen AXI DMA Channel is not halted (DMASR.Halted = 0), a write by the CPU to the TAILDESC_PTR register causes th
e AXI DMA SG Engine to start fetching descriptors or restart if it was idle (DMASR.Idle = 1). If it was not idle, wri
ting TAILDESC_PTR has no effect except to reposition the pause point.\nIf the AXI DMA Channel is halted (DMASR.Halted
= 1 and DMACR.RS = 0), a write by the CPU to the TAILDESC_PTR register has no effect except to reposition the pause p
oint.\nNote: The software must not move the tail pointer to a location that has not been updated. The software proces
ses and reallocates all completed descriptors (Cmplted = 1), clears the completed bits and then moves the tail pointe
r. The software must move the pointer to the last descriptor it updated. Descriptors must be 16-word aligned, that i
s, 0x00, 0x40, 0x80, and so forth. Any other alignment has undefined results. \n'}}},
   'S2MM_DA': {'address_offset': 72,
    'size': 32,
    'access': 'read-write',
    'description': 'S2MM DMA Destination Address Register',
    'fields': {'Destination_Address': {'bit_offset': 0,
      'bit_width': 32,
      'access': 'read-write',
      'description': 'Indicates the destination address the AXI DMA writes to transfer data from AXI4-Stream on S2MM
Channel.\nNote: If Data Realignment Engine is included, the Destination Address can be at any byte offset. If Data Re
alignment Engine is not included, the Destination Address must be S2MM Memory Map data width aligned.\n'}}},
   'S2MM_DA_MSB': {'address_offset': 76,
    'size': 32,
    'access': 'read-write',
    'description': 'S2MM Destination Address Register',
    'fields': {'Destination_Address': {'bit_offset': 0,
      'bit_width': 32,
      'access': 'read-write',
      'description': 'Indicates the MSB 32 bits of the Destination address AXI DMA writes to transfer data from AXI4-
Stream on the S2MM Channel.\nNote: If Data Realignment Engine is included, the Destination Address can be at any byte
offset. If Data Realignment Engine is not included, the Dstination Address must be S2MM Memory Map data width aligne
d.\n'}}},
   'S2MM_LENGTH': {'address_offset': 88,
    'size': 32,
```

```
    'access': 'read-write',
    'description': 'S2MM DMA Transfer Length Register',
    'fields': {'Length': {'bit_offset': 0,
      'bit_width': 26,
      'access': 'read-write',
      'description': 'Indicates the length in bytes of the S2MM buffer available to write receive data from the S2MM
channel. Writing a non-zero value to this register enables S2MM channel to receive packet data.\nAt the completion of
the S2MM transfer, the number of actual bytes written on the S2MM AXI4 interface is updated to the S2MM_LENGTH regist
er.\nNote: This value must be greater than or equal to the largest expected packet to be received on S2MM AXI4-Strea
m. Values smaller than the received packet result in undefined behavior. \n'}}}},
  'driver': pynq.lib.dma.DMA,
  'device': <pynq.pl_server.embedded_device.EmbeddedDevice at 0xffffb39cfa60>,
  'state': None,
  'bdtype': None,
  'phys_addr': 2684354560,
  'addr_range': 65536,
  'fullpath': 'axi_dma_0'},
 'example0_0': {'type': 'jie:hls:example0:1.0',
  'mem_id': 's_axi_control',
  'memtype': 'REGISTER',
  'gpio': {},
  'interrupts': {},
  'parameters': {'C_S_AXI_CONTROL_ADDR_WIDTH': '4',
   'C_S_AXI_CONTROL_DATA_WIDTH': '32',
   'Component_Name': 'design_1_example0_0_1',
   'clk_period': '10',
   'machine': '64',
   'combinational': '0',
   'latency': 'undef',
   'II': 'x',
  'EDK_IPTYPE': 'PERIPHERAL',
  'C_S_AXI_CONTROL_BASEADDR': '0xB0000000',
  'C_S_AXI_CONTROL_HIGHADDR': '0xB000FFFF',
  'ADDR_WIDTH': '4',
  'DATA_WIDTH': '32',
  'PROTOCOL': 'AXI4LITE',
  'READ_WRITE_MODE': 'READ_WRITE',
  'FREQ_HZ': '99999001',
  'ID_WIDTH': '0',
  'AWUSER_WIDTH': '0',
  'ARUSER_WIDTH': '0',
  'WUSER_WIDTH': '0',
  'RUSER_WIDTH': '0',
  'BUSER_WIDTH': '0',
  'HAS_BURST': '0',
  'HAS_LOCK': '0',
  'HAS_PROT': '0',
  'HAS_CACHE': '0',
  'HAS_QOS': '0',
  'HAS_REGION': '0',
  'HAS_WSTRB': '1',
  'HAS_BRESP': '1',
  'HAS_RRESP': '1',
  'SUPPORTS_NARROW_BURST': '0',
  'NUM_READ_OUTSTANDING': '1',
  'NUM_WRITE_OUTSTANDING': '1',
  'MAX_BURST_LENGTH': '1',
  'PHASE': '0.0',
  'CLK_DOMAIN': 'design_1_zynq_ultra_ps_e_0_1_pl_clk0',
  'NUM_READ_THREADS': '1',
  'NUM_WRITE_THREADS': '1',
  'RUSER_BITS_PER_BYTE': '0',
  'WUSER_BITS_PER_BYTE': '0',
  'INSERT_VIP': '0',
  'TDATA_NUM_BYTES': '4',
  'TDEST_WIDTH': '6',
  'TID_WIDTH': '5',
  'TUSER_WIDTH': '2',
  'LAYERED_METADATA': None,
  'HAS_TREADY': '1',
  'HAS_TSTRB': '1',
  'HAS_TKEEP': '1',
  'HAS_TLAST': '1'},
 'registers': {'CTRL': {'address_offset': 0,
   'size': 32,
   'access': 'read-write',
   'description': 'Control signals',
   'fields': {'AP_START': {'bit_offset': 0,
     'bit_width': 1,
     'access': 'read-write',
     'description': "Control signal Register for 'ap_start'."},
    'AP_DONE': {'bit_offset': 1,
     'bit_width': 1,
     'access': 'read-only',
     'description': "Control signal Register for 'ap_done'."},
    'AP_IDLE': {'bit_offset': 2,
     'bit_width': 1,
     'access': 'read-only',
     'description': "Control signal Register for 'ap_idle'."},
    'AP_READY': {'bit_offset': 3,
     'bit_width': 1,
     'access': 'read-only',
     'description': "Control signal Register for 'ap_ready'."},
```

```
      'RESERVED_1': {'bit_offset': 4,
       'bit_width': 3,
       'access': 'read-only',
       'description': 'Reserved.  0s on read.'},
      'AUTO_RESTART': {'bit_offset': 7,
       'bit_width': 1,
       'access': 'read-write',
       'description': "Control signal Register for 'auto_restart'."},
      'RESERVED_2': {'bit_offset': 8,
       'bit_width': 1,
       'access': 'read-only',
       'description': 'Reserved.  0s on read.'},
      'INTERRUPT': {'bit_offset': 9,
       'bit_width': 1,
       'access': 'read-only',
       'description': "Control signal Register for 'interrupt'."},
      'RESERVED_3': {'bit_offset': 10,
       'bit_width': 22,
       'access': 'read-only',
       'description': 'Reserved.  0s on read.'}}},
   'GIER': {'address_offset': 4,
    'size': 32,
    'access': 'read-write',
    'description': 'Global Interrupt Enable Register',
    'fields': {'Enable': {'bit_offset': 0,
      'bit_width': 1,
      'access': 'read-write',
      'description': 'Master enable for the device interrupt output to the system interrupt controller: 0 = Disabled,
1 = Enabled'},
     'RESERVED': {'bit_offset': 1,
      'bit_width': 31,
      'access': 'read-only',
      'description': 'Reserved.  0s on read.'}}},
  'IP_IER': {'address_offset': 8,
   'size': 32,
   'access': 'read-write',
   'description': 'IP Interrupt Enable Register',
   'fields': {'CHAN0_INT_EN': {'bit_offset': 0,
     'bit_width': 1,
     'access': 'read-write',
     'description': 'Enable Channel 0 (ap_done) Interrupt.  0 = Disabled, 1 = Enabled.'},
    'CHAN1_INT_EN': {'bit_offset': 1,
     'bit_width': 1,
     'access': 'read-write',
     'description': 'Enable Channel 1 (ap_ready) Interrupt.  0 = Disabled, 1 = Enabled.'},
    'RESERVED_0': {'bit_offset': 2,
     'bit_width': 30,
     'access': 'read-only',
     'description': 'Reserved.  0s on read.'}}},
  'IP_ISR': {'address_offset': 12,
   'size': 32,
   'access': 'read-write',
   'description': 'IP Interrupt Status Register',
   'fields': {'CHAN0_INT_ST': {'bit_offset': 0,
     'bit_width': 1,
     'access': 'read-only',
     'description': 'Channel 0 (ap_done) Interrupt Status. 0 = No Channel 0 interrupt, 1 = Channel 0 interrupt.'},
    'CHAN1_INT_ST': {'bit_offset': 1,
     'bit_width': 1,
     'access': 'read-only',
     'description': 'Channel 1 (ap_ready) Interrupt Status. 0 = No Channel 1 interrupt, 1 = Channel 1 interrupt.'},
    'RESERVED_0': {'bit_offset': 2,
     'bit_width': 30,
     'access': 'read-only',
     'description': 'Reserved.  0s on read.'}}}},
  'driver': pynq.overlay.DefaultIP,
  'device': <pynq.pl_server.embedded_device.EmbeddedDevice at 0xffffb39cfa60>,
  'state': None,
  'bdtype': None,
  'phys_addr': 2952790016,
  'addr_range': 65536,
  'fullpath': 'example0_0'},
 'zynq_ultra_ps_e_0': {'type': 'xilinx.com:ip:zynq_ultra_ps_e:3.4',
  'gpio': {},
  'interrupts': {},
  'parameters': {'C_DP_USE_AUDIO': '0',
   'C_DP_USE_VIDEO': '0',
   'C_MAXIGP0_DATA_WIDTH': '128',
   'C_MAXIGP1_DATA_WIDTH': '128',
   'C_MAXIGP2_DATA_WIDTH': '64',
   'C_SAXIGP0_DATA_WIDTH': '128',
   'C_SAXIGP1_DATA_WIDTH': '128',
   'C_SAXIGP2_DATA_WIDTH': '64',
   'C_SAXIGP3_DATA_WIDTH': '128',
   'C_SAXIGP4_DATA_WIDTH': '64',
   'C_SAXIGP5_DATA_WIDTH': '128',
   'C_SAXIGP6_DATA_WIDTH': '128',
   'C_USE_DIFF_RW_CLK_GP0': '0',
   'C_USE_DIFF_RW_CLK_GP1': '0',
   'C_USE_DIFF_RW_CLK_GP2': '0',
   'C_USE_DIFF_RW_CLK_GP3': '0',
   'C_USE_DIFF_RW_CLK_GP4': '0',
   'C_USE_DIFF_RW_CLK_GP5': '0',
```

```
'C_USE_DIFF_RW_CLK_GP6': '0',
'C_EN_FIFO_ENET0': '0',
'C_EN_FIFO_ENET1': '0',
'C_EN_FIFO_ENET2': '0',
'C_EN_FIFO_ENET3': '0',
'C_PL_CLK0_BUF': 'TRUE',
'C_PL_CLK1_BUF': 'FALSE',
'C_PL_CLK2_BUF': 'FALSE',
'C_PL_CLK3_BUF': 'FALSE',
'C_TRACE_PIPELINE_WIDTH': '8',
'C_EN_EMIO_TRACE': '0',
'C_TRACE_DATA_WIDTH': '32',
'C_USE_DEBUG_TEST': '0',
'C_SD0_INTERNAL_BUS_WIDTH': '5',
'C_SD1_INTERNAL_BUS_WIDTH': '5',
'C_NUM_F2P_0_INTR_INPUTS': '1',
'C_NUM_F2P_1_INTR_INPUTS': '1',
'C_EMIO_GPIO_WIDTH': '1',
'C_NUM_FABRIC_RESETS': '1',
'PSU_VALUE_SILVERSION': '3',
'PSU__USE__DDR_INTF_REQUESTED': '0',
'PSU__EN_AXI_STATUS_PORTS': '0',
'PSU__PSS_REF_CLK__FREQMHZ': '33.333',
'PSU__PSS_ALT_REF_CLK__FREQMHZ': '33.333',
'PSU__VIDEO_REF_CLK__FREQMHZ': '33.333',
'PSU__AUX_REF_CLK__FREQMHZ': '33.333',
'PSU__GT_REF_CLK__FREQMHZ': '33.333',
'PSU__VIDEO_REF_CLK__ENABLE': '0',
'PSU__VIDEO_REF_CLK__IO': '<Select>',
'PSU__PSS_ALT_REF_CLK__ENABLE': '0',
'PSU__PSS_ALT_REF_CLK__IO': '<Select>',
'PSU__CAN0__PERIPHERAL__ENABLE': '0',
'PSU__CAN0__PERIPHERAL__IO': '<Select>',
'PSU__CAN0__GRP_CLK__ENABLE': '0',
'PSU__CAN0__GRP_CLK__IO': '<Select>',
'PSU__CAN1__PERIPHERAL__ENABLE': '0',
'PSU__CAN1__PERIPHERAL__IO': '<Select>',
'PSU__CAN1__GRP_CLK__ENABLE': '0',
'PSU__CAN1__GRP_CLK__IO': '<Select>',
'PSU__CAN0_LOOP_CAN1__ENABLE': '0',
'PSU__DPAUX__PERIPHERAL__ENABLE': '0',
'PSU__DPAUX__PERIPHERAL__IO': '<Select>',
'PSU__ENET0__GRP_MDIO__ENABLE': '0',
'PSU__ACT_DDR_FREQ_MHZ': '1066.656006',
'PSU__ENET0__GRP_MDIO__IO': '<Select>',
'PSU__GEM__TSU__ENABLE': '0',
'PSU__GEM__TSU__IO': '<Select>',
'PSU__ENET0__PERIPHERAL__ENABLE': '0',
'PSU__ENET0__FIFO__ENABLE': '0',
'PSU__ENET0__PTP__ENABLE': '0',
'PSU__ENET0__PERIPHERAL__IO': '<Select>',
'PSU__ENET1__PERIPHERAL__ENABLE': '0',
'PSU__ENET1__FIFO__ENABLE': '0',
'PSU__ENET1__PTP__ENABLE': '0',
'PSU__ENET1__PERIPHERAL__IO': '<Select>',
'PSU__ENET1__GRP_MDIO__ENABLE': '0',
'PSU__FPGA_PL0_ENABLE': '1',
'PSU__FPGA_PL1_ENABLE': '1',
'PSU__FPGA_PL2_ENABLE': '0',
'PSU__FPGA_PL3_ENABLE': '0',
'PSU__ENET1__GRP_MDIO__IO': '<Select>',
'PSU__ENET2__PERIPHERAL__ENABLE': '0',
'PSU__ENET2__FIFO__ENABLE': '0',
'PSU__ENET2__PTP__ENABLE': '0',
'PSU__ENET2__PERIPHERAL__IO': '<Select>',
'PSU__ENET2__GRP_MDIO__ENABLE': '0',
'PSU__ENET2__GRP_MDIO__IO': '<Select>',
'PSU__ENET3__PERIPHERAL__ENABLE': '0',
'PSU__ENET3__FIFO__ENABLE': '0',
'PSU__ENET3__PTP__ENABLE': '0',
'PSU__ENET3__PERIPHERAL__IO': '<Select>',
'PSU__ENET3__GRP_MDIO__ENABLE': '0',
'PSU__ENET3__GRP_MDIO__IO': '<Select>',
'PSU__GPIO_EMIO__PERIPHERAL__ENABLE': '0',
'PSU__GPIO_EMIO__PERIPHERAL__IO': '<Select>',
'PSU__GPIO0_MIO__PERIPHERAL__ENABLE': '1',
'PSU__GPIO0_MIO__IO': 'MIO 0 .. 25',
'PSU__GPIO1_MIO__PERIPHERAL__ENABLE': '1',
'PSU__GPIO1_MIO__IO': 'MIO 26 .. 51',
'PSU__GPIO2_MIO__PERIPHERAL__ENABLE': '0',
'PSU__GPIO2_MIO__IO': '<Select>',
'PSU__I2C0__PERIPHERAL__ENABLE': '0',
'PSU__I2C0__PERIPHERAL__IO': '<Select>',
'PSU__I2C0__GRP_INT__ENABLE': '0',
'PSU__I2C0__GRP_INT__IO': '<Select>',
'PSU__I2C1__PERIPHERAL__ENABLE': '1',
'PSU__I2C1__PERIPHERAL__IO': 'MIO 24 .. 25',
'PSU__I2C1__GRP_INT__ENABLE': '0',
'PSU__I2C1__GRP_INT__IO': '<Select>',
'PSU__I2C0_LOOP_I2C1__ENABLE': '0',
'PSU__TESTSCAN__PERIPHERAL__ENABLE': '0',
'PSU__PCIE__PERIPHERAL__ENABLE': '0',
'PSU__PCIE__PERIPHERAL__ENDPOINT_ENABLE': '1',
```

```
'PSU__PCIE__PERIPHERAL__ROOTPORT_ENABLE': '0',
'PSU__PCIE__PERIPHERAL__ENDPOINT_IO': '<Select>',
'PSU__PCIE__PERIPHERAL__ROOTPORT_IO': '<Select>',
'PSU__PCIE__LANE0__ENABLE': '0',
'PSU__PCIE__LANE0__IO': '<Select>',
'PSU__PCIE__LANE1__ENABLE': '0',
'PSU__PCIE__LANE1__IO': '<Select>',
'PSU__PCIE__LANE2__ENABLE': '0',
'PSU__PCIE__LANE2__IO': '<Select>',
'PSU__PCIE__LANE3__ENABLE': '0',
'PSU__PCIE__LANE3__IO': '<Select>',
'PSU__PCIE__RESET__POLARITY': 'Active Low',
'PSU__GT__LINK_SPEED': '<Select>',
'PSU__GT__VLT_SWNG_LVL_4': None,
'PSU__GT__PRE_EMPH_LVL_4': None,
'PSU__USB0__REF_CLK_SEL': '<Select>',
'PSU__USB0__REF_CLK_FREQ': '<Select>',
'PSU__USB1__REF_CLK_SEL': '<Select>',
'PSU__USB1__REF_CLK_FREQ': '<Select>',
'PSU__GEM0__REF_CLK_SEL': '<Select>',
'PSU__GEM0__REF_CLK_FREQ': '<Select>',
'PSU__GEM1__REF_CLK_SEL': '<Select>',
'PSU__GEM1__REF_CLK_FREQ': '<Select>',
'PSU__GEM2__REF_CLK_SEL': '<Select>',
'PSU__GEM2__REF_CLK_FREQ': '<Select>',
'PSU__GEM3__REF_CLK_SEL': '<Select>',
'PSU__GEM3__REF_CLK_FREQ': '<Select>',
'PSU__DP__REF_CLK_SEL': '<Select>',
'PSU__DP__REF_CLK_FREQ': '<Select>',
'PSU__SATA__REF_CLK_SEL': '<Select>',
'PSU__SATA__REF_CLK_FREQ': '<Select>',
'PSU__PCIE__REF_CLK_SEL': '<Select>',
'PSU__PCIE__REF_CLK_FREQ': '<Select>',
'PSU__DP__LANE_SEL': '<Select>',
'PSU__PCIE__DEVICE_PORT_TYPE': '<Select>',
'PSU__PCIE__MAXIMUM_LINK_WIDTH': '<Select>',
'PSU__PCIE__LINK_SPEED': '<Select>',
'PSU__PCIE__INTERFACE_WIDTH': '<Select>',
'PSU__PCIE__BAR0_ENABLE': '0',
'PSU__PCIE__BAR0_TYPE': '<Select>',
'PSU__PCIE__BAR0_SCALE': '<Select>',
'PSU__PCIE__BAR0_64BIT': '0',
'PSU__PCIE__BAR0_SIZE': '<Select>',
'PSU__PCIE__BAR0_VAL': None,
'PSU__PCIE__BAR0_PREFETCHABLE': '0',
'PSU__PCIE__BAR1_ENABLE': '0',
'PSU__PCIE__BAR1_TYPE': '<Select>',
'PSU__PCIE__BAR1_SCALE': '<Select>',
'PSU__PCIE__BAR1_64BIT': '0',
'PSU__PCIE__BAR1_SIZE': '<Select>',
'PSU__PCIE__BAR1_VAL': None,
'PSU__PCIE__BAR1_PREFETCHABLE': '0',
'PSU__PCIE__BAR2_ENABLE': '0',
'PSU__PCIE__BAR2_TYPE': '<Select>',
'PSU__PCIE__BAR2_SCALE': '<Select>',
'PSU__PCIE__BAR2_64BIT': '0',
'PSU__PCIE__BAR2_SIZE': '<Select>',
'PSU__PCIE__BAR2_VAL': None,
'PSU__PCIE__BAR2_PREFETCHABLE': '0',
'PSU__PCIE__BAR3_ENABLE': '0',
'PSU__PCIE__BAR3_TYPE': '<Select>',
'PSU__PCIE__BAR3_SCALE': '<Select>',
'PSU__PCIE__BAR3_64BIT': '0',
'PSU__PCIE__BAR3_SIZE': '<Select>',
'PSU__PCIE__BAR3_VAL': None,
'PSU__PCIE__BAR3_PREFETCHABLE': '0',
'PSU__PCIE__BAR4_ENABLE': '0',
'PSU__PCIE__BAR4_TYPE': '<Select>',
'PSU__PCIE__BAR4_SCALE': '<Select>',
'PSU__PCIE__BAR4_64BIT': '0',
'PSU__PCIE__BAR4_SIZE': '<Select>',
'PSU__PCIE__BAR4_VAL': None,
'PSU__PCIE__BAR4_PREFETCHABLE': '0',
'PSU__PCIE__BAR5_ENABLE': '0',
'PSU__PCIE__BAR5_TYPE': '<Select>',
'PSU__PCIE__BAR5_SCALE': '<Select>',
'PSU__PCIE__BAR5_64BIT': '0',
'PSU__PCIE__BAR5_SIZE': '<Select>',
'PSU__PCIE__BAR5_VAL': None,
'PSU__PCIE__BAR5_PREFETCHABLE': '0',
'PSU__PCIE__EROM_ENABLE': '0',
'PSU__PCIE__EROM_SCALE': '<Select>',
'PSU__PCIE__EROM_SIZE': '<Select>',
'PSU__PCIE__EROM_VAL': None,
'PSU__PCIE__CAP_SLOT_IMPLEMENTED': '<Select>',
'PSU__PCIE__MAX_PAYLOAD_SIZE': '<Select>',
'PSU__PCIE__LEGACY_INTERRUPT': '<Select>',
'PSU__PCIE__VENDOR_ID': None,
'PSU__PCIE__DEVICE_ID': None,
'PSU__PCIE__REVISION_ID': None,
'PSU__PCIE__SUBSYSTEM_VENDOR_ID': None,
'PSU__PCIE__SUBSYSTEM_ID': None,
'PSU__PCIE__BASE_CLASS_MENU': '<Select>',
```

```
'PSU__PCIE__USE_CLASS_CODE_LOOKUP_ASSISTANT': '<Select>',
'PSU__PCIE__SUB_CLASS_INTERFACE_MENU': '<Select>',
'PSU__PCIE__CLASS_CODE_BASE': None,
'PSU__PCIE__CLASS_CODE_SUB': None,
'PSU__PCIE__CLASS_CODE_INTERFACE': None,
'PSU__PCIE__CLASS_CODE_VALUE': None,
'PSU__PCIE__AER_CAPABILITY': '0',
'PSU__PCIE__CORRECTABLE_INT_ERR': '0',
'PSU__PCIE__HEADER_LOG_OVERFLOW': '0',
'PSU__PCIE__RECEIVER_ERR': '0',
'PSU__PCIE__SURPRISE_DOWN': '0',
'PSU__PCIE__FLOW_CONTROL_ERR': '0',
'PSU__PCIE__COMPLTION_TIMEOUT': '0',
'PSU__PCIE__COMPLETER_ABORT': '0',
'PSU__PCIE__RECEIVER_OVERFLOW': '0',
'PSU__PCIE__ECRC_ERR': '0',
'PSU__PCIE__ACS_VIOLAION': '0',
'PSU__PCIE__UNCORRECTABL_INT_ERR': '0',
'PSU__PCIE__MC_BLOCKED_TLP': '0',
'PSU__PCIE__ATOMICOP_EGRESS_BLOCKED': '0',
'PSU__PCIE__TLP_PREFIX_BLOCKED': '0',
'PSU__PCIE__FLOW_CONTROL_PROTOCOL_ERR': '0',
'PSU__PCIE__ACS_VIOLATION': '0',
'PSU__PCIE__MULTIHEADER': '0',
'PSU__PCIE__ECRC_CHECK': '0',
'PSU__PCIE__ECRC_GEN': '0',
'PSU__PCIE__PERM_ROOT_ERR_UPDATE': '0',
'PSU__PCIE__CRS_SW_VISIBILITY': '0',
'PSU__PCIE__INTX_GENERATION': '0',
'PSU__PCIE__INTX_PIN': '<Select>',
'PSU__PCIE__MSI_CAPABILITY': '0',
'PSU__PCIE__MSI_64BIT_ADDR_CAPABLE': '0',
'PSU__PCIE__MSI_MULTIPLE_MSG_CAPABLE': '<Select>',
'PSU__PCIE__MSIX_CAPABILITY': '0',
'PSU__PCIE__MSIX_TABLE_SIZE': '0',
'PSU__PCIE__MSIX_TABLE_OFFSET': '0',
'PSU__PCIE__MSIX_BAR_INDICATOR': None,
'PSU__PCIE__MSIX_PBA_OFFSET': '0',
'PSU__PCIE__MSIX_PBA_BAR_INDICATOR': None,
'PSU__PCIE__BRIDGE_BAR_INDICATOR': '<Select>',
'PSU_IMPORT_BOARD_PRESET': None,
'PSU__PROTECTION__SUBSYSTEMS': 'PMU Firmware:PMU|Secure Subsystem:',
'PSU__PROTECTION__MASTERS_TZ': 'GEM0:NonSecure|SD1:NonSecure|GEM2:NonSecure|GEM1:NonSecure|GEM3:NonSecure|PCIe:NonSecure|DP:NonSecure|NAND:NonSecure|GPU:NonSecure|USB1:NonSecure|USB0:NonSecure|LDMA:NonSecure|FDMA:NonSecure|QSPI:NonSecure|SD0:NonSecure',
'PSU__PROTECTION__MASTERS': 'USB1:NonSecure;0|USB0:NonSecure;0|S_AXI_LPD:NA;0|S_AXI_HPC1_FPD:NA;0|S_AXI_HPC0_FPD:NA;0|S_AXI_HP3_FPD:NA;0|S_AXI_HP2_FPD:NA;1|S_AXI_HP1_FPD:NA;0|S_AXI_HP0_FPD:NA;1|S_AXI_ACP:NA;0|S_AXI_ACE:NA;0|SD1:NonSecure;0|SD0:NonSecure;0|SATA1:NonSecure;0|SATA0:NonSecure;0|RPU1:Secure;1|RPU0:Secure;1|QSPI:NonSecure;1|PMU:NA;1|PCIe:NonSecure;0|NAND:NonSecure;0|LDMA:NonSecure;1|GPU:NonSecure;1|GEM3:NonSecure;0|GEM2:NonSecure;0|GEM1:NonSecure;0|GEM0:NonSecure;0|FDMA:NonSecure;1|DP:NonSecure;0|DAP:NA;1|Coresight:NA;1|CSU:NA;1|APU:NA;1',
'PSU__PROTECTION__DDR_SEGMENTS': 'NONE',
'PSU__PROTECTION__OCM_SEGMENTS': 'NONE',
'PSU__PROTECTION__LPD_SEGMENTS': 'SA:0xFF980000; SIZE:64; UNIT:KB; RegionTZ:Secure; WrAllowed:Read/Write; subsystemId:PMU Firmware| SA:0xFF5E0000; SIZE:2560; UNIT:KB; RegionTZ:Secure; WrAllowed:Read/Write; subsystemId:PMU Firmware| SA:0xFFCC0000; SIZE:64; UNIT:KB; RegionTZ:Secure; WrAllowed:Read/Write; subsystemId:PMU Firmware| SA:0xFF180000; SIZE:768; UNIT:KB; RegionTZ:Secure; WrAllowed:Read/Write; subsystemId:PMU Firmware| SA:0xFF410000; SIZE:640; UNIT:KB; RegionTZ:Secure; WrAllowed:Read/Write; subsystemId:PMU Firmware| SA:0xFFA70000; SIZE:64; UNIT:KB; RegionTZ:Secure; WrAllowed:Read/Write; subsystemId:PMU Firmware| SA:0xFF9A0000; SIZE:64; UNIT:KB; RegionTZ:Secure; WrAllowed:Read/Write; subsystemId:PMU Firmware|SA:0xFF5E0000 ; SIZE:2560; UNIT:KB; RegionTZ:Secure ; WrAllowed:Read/Write; subsystemId:Secure Subsystem|SA:0xFFCC0000 ; SIZE:64; UNIT:KB; RegionTZ:Secure ; WrAllowed:Read/Write; subsystemId:Secure Subsystem|SA:0xFF180000 ; SIZE:768; UNIT:KB; RegionTZ:Secure ; WrAllowed:Read/Write; subsystemId:Secure Subsystem|SA:0xFF9A0000 ; SIZE:64; UNIT:KB; RegionTZ:Secure ; WrAllowed:Read/Write; subsystemId:Secure Subsystem',
'PSU__PROTECTION__FPD_SEGMENTS': 'SA:0xFD1A0000; SIZE:1280; UNIT:KB; RegionTZ:Secure; WrAllowed:Read/Write; subsystemId:PMU Firmware| SA:0xFD000000; SIZE:64; UNIT:KB; RegionTZ:Secure; WrAllowed:Read/Write; subsystemId:PMU Firmware| SA:0xFD010000; SIZE:64; UNIT:KB; RegionTZ:Secure; WrAllowed:Read/Write; subsystemId:PMU Firmware| SA:0xFD020000; SIZE:64; UNIT:KB; RegionTZ:Secure; WrAllowed:Read/Write; subsystemId:PMU Firmware| SA:0xFD030000; SIZE:64; UNIT:KB; RegionTZ:Secure; WrAllowed:Read/Write; subsystemId:PMU Firmware| SA:0xFD040000; SIZE:64; UNIT:KB; RegionTZ:Secure; WrAllowed:Read/Write; subsystemId:PMU Firmware| SA:0xFD050000; SIZE:64; UNIT:KB; RegionTZ:Secure; WrAllowed:Read/Write; subsystemId:PMU Firmware| SA:0xFD610000; SIZE:512; UNIT:KB; RegionTZ:Secure; WrAllowed:Read/Write; subsystemId:PMU Firmware| SA:0xFD5D0000; SIZE:64; UNIT:KB; RegionTZ:Secure; WrAllowed:Read/Write; subsystemId:PMU Firmware|SA:0xFD1A0000 ; SIZE:1280; UNIT:KB; RegionTZ:Secure ; WrAllowed:Read/Write; subsystemId:Secure Subsystem',
'PSU__PROTECTION__DEBUG': '0',
'PSU__PROTECTION__SLAVES': 'LPD;USB3_1_XHCI;FE300000;FE3FFFFF;0|LPD;USB3_1;FF9E0000;FF9EFFFF;0|LPD;USB3_0_XHCI;FE200000;FE2FFFFF;0|LPD;USB3_0;FF9D0000;FF9DFFFF;0|LPD;UART1;FF010000;FF01FFFF;0|LPD;UART0;FF000000;FF00FFFF;0|LPD;TTC3;FF140000;FF14FFFF;1|LPD;TTC2;FF130000;FF13FFFF;1|LPD;TTC1;FF120000;FF12FFFF;1|LPD;TTC0;FF110000;FF11FFFF;1|FPD;SWDT1;FD4D0000;FD4DFFFF;1|LPD;SWDT0;FF150000;FF15FFFF;1|LPD;SPI1;FF050000;FF05FFFF;1|LPD;SPI0;FF040000;FF04FFFF;0|FPD;SMMU_REG;FD5F0000;FD5FFFFF;1|FPD;SMMU;FD800000;FDFFFFFF;1|FPD;SIOU;FD3D0000;FD3DFFFF;1|FPD;SERDES;FD400000;FD47FFFF;1|LPD;SD1;FF170000;FF17FFFF;0|LPD;SD0;FF160000;FF16FFFF;0|FPD;SATA;FD0C0000;FD0CFFFF;0|LPD;RTC;FFA60000;FFA6FFFF;1|LPD;RSA_CORE;FFCE0000;FFCEFFFF;1|LPD;RPU;FF9A0000;FF9AFFFF;1|LPD;R5_TCM_RAM_GLOBAL;FFE00000;FFE3FFFF;1|LPD;R5_1_Instruction_Cache;FFEC0000;FFECFFFF;1|LPD;R5_1_Data_Cache;FFED0000;FFEDFFFF;1|LPD;R5_1_BTCM_GLOBAL;FFEB0000;FFEBFFFF;1|LPD;R5_1_ATCM_GLOBAL;FFE90000;FFE9FFFF;1|LPD;R5_0_Instruction_Cache;FFE40000;FFE4FFFF;1|LPD;R5_0_Data_Cache;FFE50000;FFE5FFFF;1|LPD;R5_0_BTCM_GLOBAL;FFE20000;FFE2FFFF;1|LPD;R5_0_ATCM_GLOBAL;FFE00000;FFE0FFFF;1|LPD;QSPI_Linear_Address;C0000000;DFFFFFFF;1|LPD;QSPI;FF0F0000;FF0FFFFF;1|LPD;PMU_RAM;FFDC0000;FFDDFFFF;1|LPD;PMU_GLOBAL;FFD80000;FFDBFFFF;1|FPD;PCIE_MAIN;FD0E0000;FD0EFFFF;0|FPD;PCIE_LOW;E0000000;EFFFFFFF;0|FPD;PCIE_HIGH2;8000000000;BFFFFFFFFF;0|FPD;PCIE_HIGH1;600000000;7FFFFFFF;0|FPD;PCIE_DMA;FD0F0000;FD0FFFFF;0|FPD;PCIE_ATTRIB;FD480000;FD48FFFF;0|LPD;OCM_XMPU_CFG;FFA70000;FFA7FFFF;1|LPD;OCM_SLCR;FF960000;FF96FFFF;1|OCM;OCM;FFFC0000;FFFFFFFF;1|LPD;NAND;FF100000;FF10FFFF;0|LPD;MBISTJTAG;FFCF0000;FFCFFFFF;1|LPD;LPD_XPPU_SINK;FF9C0000;FF9CFFFF;1|LPD;LPD_XPPU;FF980000;FF98FFFF;1|LPD;LPD_SLCR_SECURE;FF4B0000;FF4DFFFF;1|LPD;LPD_SLCR;FF410000;FF4AFFFF;1|LPD;LPD_GPV;FE100000;FE1FFFFF;1|LPD;LPD_DMA_7;FFAF0000;FFAFFFFF;1|LPD;LPD_DMA_6;FFAE0000;FFAEFFFF;1|LPD;LPD_DMA_5;FFAD0000;FFADFFFF;1|LPD;LPD_DMA_4;FFAC0000;FFACFFFF;1|LPD;LPD_DMA_3;FFAB0000;FFABFFFF;1|LPD;LPD_DMA_2;FFAA0000;FFAAFFFF;1|LPD;LPD_DMA_1;FFA90000;FFA9FFFF;1|LPD;LPD_DMA_0;FFA80000;FFA8FFFF;1|LPD;IPI_CTRL;FF380000;FF3FFFFF;1|LPD;IOU_SLCR;FF180000;FF23FFFF;1|LPD;IOU_SECURE_SLCR;FF240000;FF24FFFF;1|LPD;IOU_SCNTRS;FF260000;FF26FFFF;1|LPD;IOU_SCNTR;FF250000;FF25FFFF;1|LPD;IOU_GPV;FE000000;FE0FFFFF;1|LPD;I2C1;FF030000;FF03FFFF;1|LPD;I2C0;FF020000;FF02FFFF;0|FPD;GPU;FD4B0000;FD4BFFFF;1|LPD;GPIO;FF0A0000;FF0AFFFF;1|LPD;GEM3;FF0E0000;FF0EFFFF;0|LPD;GEM
```

```
2;FF0D0000;FF0DFFFF;0|LPD;GEM1;FF0C0000;FF0CFFFF;0|LPD;GEM0;FF0B0000;FF0BFFFF;0|FPD;FPD_XMPU_SINK;FD4F0000;FD4FFFFF;1
|FPD;FPD_XMPU_CFG;FD5D0000;FD5DFFFF;1|FPD;FPD_SLCR_SECURE;FD690000;FD6CFFFF;1|FPD;FPD_SLCR;FD610000;FD68FFFF;1|FPD;FP
D_DMA_CH7;FD570000;FD57FFFF;1|FPD;FPD_DMA_CH6;FD560000;FD56FFFF;1|FPD;FPD_DMA_CH5;FD550000;FD55FFFF;1|FPD;FPD_DMA_CH
4;FD540000;FD54FFFF;1|FPD;FPD_DMA_CH3;FD530000;FD53FFFF;1|FPD;FPD_DMA_CH2;FD520000;FD52FFFF;1|FPD;FPD_DMA_CH1;FD51000
0;FD51FFFF;1|FPD;FPD_DMA_CH0;FD500000;FD50FFFF;1|LPD;EFUSE;FFCC0000;FFCCFFFF;1|FPD;Display Port;FD4A0000;FD4AFFFF;0|F
PD;DPDMA;FD4C0000;FD4CFFFF;0|FPD;DDR_XMPU5_CFG;FD050000;FD05FFFF;1|FPD;DDR_XMPU4_CFG;FD040000;FD04FFFF;1|FPD;DDR_XMPU
3_CFG;FD030000;FD03FFFF;1|FPD;DDR_XMPU2_CFG;FD020000;FD02FFFF;1|FPD;DDR_XMPU1_CFG;FD010000;FD01FFFF;1|FPD;DDR_XMPU0_C
FG;FD000000;FD00FFFF;1|FPD;DDR_QOS_CTRL;FD090000;FD09FFFF;1|FPD;DDR_PHY;FD080000;FD08FFFF;1|DDR;DDR_LOW;0;7FFFFFFF;1|
DDR;DDR_HIGH;800000000;87FFFFFF;1|FPD;DDDR_CTRL;FD070000;FD070FFF;1|LPD;Coresight;FE800000;FEFFFFFF;1|LPD;CSU_DMA;FF
C80000;FFC9FFFF;1|LPD;CSU;FFCA0000;FFCAFFFF;1|LPD;CRL_APB;FF5E0000;FF85FFFF;1|FPD;CRF_APB;FD1A0000;FD2DFFFF;1|FPD;CCI
_REG;FD5E0000;FD5EFFFF;1|LPD;CAN1;FF070000;FF07FFFF;0|LPD;CAN0;FF060000;FF06FFFF;0|FPD;APU;FD5C0000;FD5CFFFF;1|LPD;AP
M_INTC_IOU;FFA20000;FFA2FFFF;1|LPD;APM_FPD_LPD;FFA30000;FFA3FFFF;1|FPD;APM_5;FD490000;FD49FFFF;1|FPD;APM_0;FD0B0000;F
D0BFFFF;1|LPD;APM2;FFA10000;FFA1FFFF;1|LPD;APM1;FFA00000;FFA0FFFF;1|LPD;AMS;FFA50000;FFA5FFFF;1|FPD;AFI_5;FD3B0000;FD
3BFFFF;1|FPD;AFI_4;FD3A0000;FD3AFFFF;1|FPD;AFI_3;FD390000;FD39FFFF;1|FPD;AFI_2;FD380000;FD38FFFF;1|FPD;AFI_1;FD37000
0;FD37FFFF;1|FPD;AFI_0;FD360000;FD36FFFF;1|LPD;AFIFM6;FF9B0000;FF9BFFFF;1|FPD;ACPU_GIC;F9010000;F907FFFF;1',
   'PSU__PROTECTION__PRESUBSYSTEMS': 'NONE',
   'PSU__PROTECTION__ENABLE': '0',
   'PSU__DDR_SW_REFRESH_ENABLED': '1',
   'PSU__PROTECTION__LOCK_UNUSED_SEGMENTS': '0',
   'PSU__EP__IP': '0',
   'PSU__ACTUAL__IP': '1',
   'SUBPRESET1': 'Custom',
   'SUBPRESET2': 'Custom',
   'PSU_UIPARAM_GENERATE_SUMMARY': '<Select>',
   'PSU_MIO_TREE_PERIPHERALS': 'Quad SPI Flash#Quad SPI Flash#Quad SPI Flash#Quad SPI Flash#Quad SPI Flash#Quad SPI F
lash#SPI 1#GPIO0 MIO#GPIO0 MIO#SPI 1#SPI 1#SPI 1#GPIO0 MIO#GPIO0 MIO#GPIO0 MIO#GPIO0 MIO#GPIO0 MIO#GPIO0 MIO#GPIO0 MI
O#GPIO0 MIO#GPIO0 MIO#GPIO0 MIO#GPIO0 MIO#GPIO0 MIO#I2C 1#I2C 1#PMU GPI 0#GPIO1 MIO#GPIO1 MIO#GPIO1 MIO#GPIO1 MIO#PMU
GPI 5#PMU GPO 0#PMU GPO 1#PMU GPO 2#PMU GPO 3#GPIO1 MIO#GPIO1 MIO#GPIO1 MIO#GPIO1 MIO#GPIO1 MIO#GPIO1 MIO#GPIO1 MIO#G
PIO1 MIO#GPIO1 MIO#GPIO1 MIO#GPIO1 MIO#GPIO1 MIO#GPIO1 MIO#GPIO1 MIO#GPIO1 MIO#GPIO1 MIO#############################',
   'PSU_MIO_TREE_SIGNALS': 'sclk_out#miso_mo1#mo2#mo3#mosi_mi0#n_ss_out#sclk_out#gpio0[7]#gpio0[8]#n_ss_out[0]#miso#m
osi#gpio0[12]#gpio0[13]#gpio0[14]#gpio0[15]#gpio0[16]#gpio0[17]#gpio0[18]#gpio0[19]#gpio0[20]#gpio0[21]#gpio0[22]#gpi
o0[23]#scl_out#sda_out#gpi[0]#gpio1[27]#gpio1[28]#gpio1[29]#gpio1[30]#gpi[5]#gpo[0]#gpo[1]#gpo[2]#gpo[3]#gpio1[36]#gp
io1[37]#gpio1[38]#gpio1[39]#gpio1[40]#gpio1[41]#gpio1[42]#gpio1[43]#gpio1[44]#gpio1[45]#gpio1[46]#gpio1[47]#gpio1[48]
#gpio1[49]#gpio1[50]#gpio1[51]#########################',
   'PSU_PERIPHERAL_BOARD_PRESET': None,
   'PSU__NAND__PERIPHERAL__IO': '<Select>',
   'PSU__NAND__PERIPHERAL__ENABLE': '0',
   'PSU__NAND__READY_BUSY__ENABLE': '0',
   'PSU__NAND__READY0_BUSY__ENABLE': '0',
   'PSU__NAND__READY1_BUSY__ENABLE': '0',
   'PSU__NAND__READY_BUSY__IO': '<Select>',
   'PSU__NAND__READY0_BUSY__IO': '<Select>',
   'PSU__NAND__READY1_BUSY__IO': '<Select>',
   'PSU__NAND__CHIP_ENABLE__ENABLE': '0',
   'PSU__NAND__CHIP_ENABLE__IO': '<Select>',
   'PSU__NAND__DATA_STROBE__ENABLE': '0',
   'PSU__NAND__DATA_STROBE__IO': '<Select>',
   'PSU__PJTAG__PERIPHERAL__ENABLE': '0',
   'PSU__PJTAG__PERIPHERAL__IO': '<Select>',
   'PSU__PMU__AIBACK__ENABLE': '0',
   'PSU__PMU__PLERROR__ENABLE': '0',
   'PSU__PMU__PERIPHERAL__ENABLE': '1',
   'PSU__PMU__PERIPHERAL__IO': '<Select>',
   'PSU__PMU__EMIO_GPI__ENABLE': '0',
   'PSU__PMU__EMIO_GPO__ENABLE': '0',
   'PSU__PMU__GPI0__ENABLE': '1',
   'PSU__PMU__GPI1__ENABLE': '0',
   'PSU__PMU__GPI2__ENABLE': '0',
   'PSU__PMU__GPI3__ENABLE': '0',
   'PSU__PMU__GPI4__ENABLE': '0',
   'PSU__PMU__GPI5__ENABLE': '1',
   'PSU__PMU__GPO0__ENABLE': '1',
   'PSU__PMU__GPO1__ENABLE': '1',
   'PSU__PMU__GPO2__ENABLE': '1',
   'PSU__PMU__GPO3__ENABLE': '1',
   'PSU__PMU__GPO4__ENABLE': '0',
   'PSU__PMU__GPO5__ENABLE': '0',
   'PSU__PMU__GPI0__IO': 'MIO 26',
   'PSU__PMU__GPI1__IO': '<Select>',
   'PSU__PMU__GPI2__IO': '<Select>',
   'PSU__PMU__GPI3__IO': '<Select>',
   'PSU__PMU__GPI4__IO': '<Select>',
   'PSU__PMU__GPI5__IO': 'MIO 31',
   'PSU__PMU__GPO0__IO': 'MIO 32',
   'PSU__PMU__GPO1__IO': 'MIO 33',
   'PSU__PMU__GPO2__IO': 'MIO 34',
   'PSU__PMU__GPO3__IO': 'MIO 35',
   'PSU__PMU__GPO4__IO': '<Select>',
   'PSU__PMU__GPO5__IO': '<Select>',
   'PSU__PMU__GPO2__POLARITY': 'high',
   'PSU__PMU__GPO3__POLARITY': 'low',
   'PSU__PMU__GPO4__POLARITY': '<Select>',
   'PSU__PMU__GPO5__POLARITY': '<Select>',
   'PSU__CSU__PERIPHERAL__ENABLE': '0',
   'PSU__CSU__PERIPHERAL__IO': '<Select>',
   'PSU__QSPI__PERIPHERAL__ENABLE': '1',
   'PSU__QSPI__PERIPHERAL__IO': 'MIO 0 .. 5',
   'PSU__QSPI__PERIPHERAL__MODE': 'Single',
   'PSU__QSPI__PERIPHERAL__DATA_MODE': 'x4',
   'PSU__QSPI__GRP_FBCLK__ENABLE': '0',
   'PSU__QSPI__GRP_FBCLK__IO': '<Select>',
   'PSU__SD0__PERIPHERAL__ENABLE': '0',
   'PSU__SD0__PERIPHERAL__IO': '<Select>',
```

```
'PSU__SD0__GRP_CD__ENABLE': '0',
'PSU__SD0__GRP_CD__IO': '<Select>',
'PSU__SD0__GRP_POW__ENABLE': '0',
'PSU__SD0__GRP_POW__IO': '<Select>',
'PSU__SD0__GRP_WP__ENABLE': '0',
'PSU__SD0__GRP_WP__IO': '<Select>',
'PSU__SD0__SLOT_TYPE': '<Select>',
'PSU__SD0__RESET__ENABLE': '0',
'PSU__SD0__DATA_TRANSFER_MODE': '<Select>',
'PSU__SD0__CLK_50_SDR_ITAP_DLY': '0x00',
'PSU__SD0__CLK_50_SDR_OTAP_DLY': '0x00',
'PSU__SD0__CLK_50_DDR_ITAP_DLY': '0x00',
'PSU__SD0__CLK_50_DDR_OTAP_DLY': '0x00',
'PSU__SD0__CLK_100_SDR_OTAP_DLY': '0x00',
'PSU__SD0__CLK_200_SDR_OTAP_DLY': '0x00',
'PSU__SD1__PERIPHERAL__ENABLE': '0',
'PSU__SD1__PERIPHERAL__IO': '<Select>',
'PSU__SD1__GRP_CD__ENABLE': '0',
'PSU__SD1__GRP_CD__IO': '<Select>',
'PSU__SD1__GRP_POW__ENABLE': '0',
'PSU__SD1__GRP_POW__IO': '<Select>',
'PSU__SD1__GRP_WP__ENABLE': '0',
'PSU__SD1__GRP_WP__IO': '<Select>',
'PSU__SD1__SLOT_TYPE': '<Select>',
'PSU__SD1__RESET__ENABLE': '0',
'PSU__SD1__DATA_TRANSFER_MODE': '<Select>',
'PSU__SD1__CLK_50_SDR_ITAP_DLY': '0x00',
'PSU__SD1__CLK_50_SDR_OTAP_DLY': '0x00',
'PSU__SD1__CLK_50_DDR_ITAP_DLY': '0x00',
'PSU__SD1__CLK_50_DDR_OTAP_DLY': '0x00',
'PSU__SD1__CLK_100_SDR_OTAP_DLY': '0x00',
'PSU__SD1__CLK_200_SDR_OTAP_DLY': '0x00',
'PSU__DEVICE_TYPE': 'EV',
'PSU_SMC_CYCLE_T0': 'NA',
'PSU_SMC_CYCLE_T1': 'NA',
'PSU_SMC_CYCLE_T2': 'NA',
'PSU_SMC_CYCLE_T3': 'NA',
'PSU_SMC_CYCLE_T4': 'NA',
'PSU_SMC_CYCLE_T5': 'NA',
'PSU_SMC_CYCLE_T6': 'NA',
'PSU__SPI0__PERIPHERAL__ENABLE': '0',
'PSU__SPI0__PERIPHERAL__IO': '<Select>',
'PSU__SPI0__GRP_SS0__ENABLE': '0',
'PSU__SPI0__GRP_SS0__IO': '<Select>',
'PSU__SPI0__GRP_SS1__ENABLE': '0',
'PSU__SPI0__GRP_SS1__IO': '<Select>',
'PSU__SPI0__GRP_SS2__ENABLE': '0',
'PSU__SPI0__GRP_SS2__IO': '<Select>',
'PSU__SPI1__PERIPHERAL__ENABLE': '1',
'PSU__SPI1__PERIPHERAL__IO': 'MIO 6 .. 11',
'PSU__SPI1__GRP_SS0__ENABLE': '1',
'PSU__SPI1__GRP_SS0__IO': 'MIO 9',
'PSU__SPI1__GRP_SS1__ENABLE': '0',
'PSU__SPI1__GRP_SS1__IO': '<Select>',
'PSU__SPI1__GRP_SS2__ENABLE': '0',
'PSU__SPI1__GRP_SS2__IO': '<Select>',
'PSU__SPI0_LOOP_SPI1__ENABLE': '0',
'PSU__LPD_SLCR__CSUPMU_WDT_CLK_SEL__SELECT': 'APB',
'PSU__SWDT0__PERIPHERAL__ENABLE': '1',
'PSU__SWDT0__CLOCK__ENABLE': '0',
'PSU__SWDT0__RESET__ENABLE': '0',
'PSU__SWDT0__PERIPHERAL__IO': 'NA',
'PSU__SWDT0__CLOCK__IO': '<Select>',
'PSU__SWDT0__RESET__IO': '<Select>',
'PSU__SWDT1__PERIPHERAL__ENABLE': '1',
'PSU__SWDT1__CLOCK__ENABLE': '0',
'PSU__SWDT1__RESET__ENABLE': '0',
'PSU__SWDT1__PERIPHERAL__IO': 'NA',
'PSU__SWDT1__CLOCK__IO': '<Select>',
'PSU__SWDT1__RESET__IO': '<Select>',
'PSU__UART0__BAUD_RATE': '<Select>',
'PSU__TRACE__PERIPHERAL__ENABLE': '0',
'PSU__TRACE__PERIPHERAL__IO': '<Select>',
'PSU__TRACE__WIDTH': '<Select>',
'PSU__TRACE__INTERNAL_WIDTH': '32',
'PSU_SD0_INTERNAL_BUS_WIDTH': '8',
'PSU__TTC0__PERIPHERAL__ENABLE': '1',
'PSU__TTC0__CLOCK__ENABLE': '0',
'PSU__TTC0__WAVEOUT__ENABLE': '0',
'PSU__TTC0__CLOCK__IO': '<Select>',
'PSU__TTC0__WAVEOUT__IO': '<Select>',
'PSU__TTC0__PERIPHERAL__IO': 'NA',
'PSU__TTC1__PERIPHERAL__ENABLE': '1',
'PSU__TTC1__PERIPHERAL__IO': 'NA',
'PSU__UART1__BAUD_RATE': '<Select>',
'PSU__TTC1__CLOCK__ENABLE': '0',
'PSU__TTC1__WAVEOUT__ENABLE': '0',
'PSU__TTC1__CLOCK__IO': '<Select>',
'PSU__TTC1__WAVEOUT__IO': '<Select>',
'PSU__TTC2__PERIPHERAL__ENABLE': '1',
'PSU__TTC2__PERIPHERAL__IO': 'NA',
'PSU__TTC2__CLOCK__ENABLE': '0',
'PSU__TTC2__WAVEOUT__ENABLE': '0',
```

```
'PSU__TTC2__CLOCK__IO': '<Select>',
'PSU__TTC2__WAVEOUT__IO': '<Select>',
'PSU__TTC3__PERIPHERAL__ENABLE': '1',
'PSU__TTC3__PERIPHERAL__IO': 'NA',
'PSU__TTC3__CLOCK__ENABLE': '0',
'PSU__TTC3__WAVEOUT__ENABLE': '0',
'PSU__TTC3__CLOCK__IO': '<Select>',
'PSU__TTC3__WAVEOUT__IO': '<Select>',
'PSU__CSUPMU__PERIPHERAL__VALID': '1',
'PSU__DDRC__AL': '0',
'PSU__DDRC__BANK_ADDR_COUNT': '2',
'PSU__DDRC__BUS_WIDTH': '64 Bit',
'PSU__DDRC__CL': '16',
'PSU__DDRC__CLOCK_STOP_EN': '0',
'PSU_DYNAMIC_DDR_CONFIG_EN': '0',
'PSU__DDRC__COL_ADDR_COUNT': '10',
'PSU__DDRC__RANK_ADDR_COUNT': '0',
'PSU__DDRC__CWL': '14',
'PSU__DDRC__BG_ADDR_COUNT': '1',
'PSU__DDRC__DEVICE_CAPACITY': '8192 MBits',
'PSU__DDRC__DRAM_WIDTH': '16 Bits',
'PSU__DDRC__ECC': 'Disabled',
'PSU__DDRC__ECC_SCRUB': '0',
'PSU__DDRC__ENABLE': '1',
'PSU__DDRC__FREQ_MHZ': '1',
'PSU__DDRC__HIGH_TEMP': '<Select>',
'PSU__DDRC__MEMORY_TYPE': 'DDR 4',
'PSU__DDRC__PARTNO': '<Select>',
'PSU__DDRC__ROW_ADDR_COUNT': '16',
'PSU__DDRC__SPEED_BIN': 'DDR4_2400R',
'PSU__DDRC__T_FAW': '30.0',
'PSU__DDRC__T_RAS_MIN': '33',
'PSU__DDRC__T_RC': '47.06',
'PSU__DDRC__T_RCD': '16',
'PSU__DDRC__T_RP': '16',
'PSU__DDRC__TRAIN_DATA_EYE': '1',
'PSU__DDRC__TRAIN_READ_GATE': '1',
'PSU__DDRC__TRAIN_WRITE_LEVEL': '1',
'PSU__DDRC__VREF': '1',
'PSU__DDRC__VIDEO_BUFFER_SIZE': '0',
'PSU__DDRC__BRC_MAPPING': 'ROW_BANK_COL',
'PSU__DDRC__DIMM_ADDR_MIRROR': '0',
'PSU__DDRC__STATIC_RD_MODE': '0',
'PSU__DDRC__DDR4_MAXPWR_SAVING_EN': '0',
'PSU__DDRC__PWR_DOWN_EN': '0',
'PSU__DDRC__DEEP_PWR_DOWN_EN': '0',
'PSU__DDRC__PLL_BYPASS': '0',
'PSU__DDRC__DDR4_T_REF_MODE': '0',
'PSU__DDRC__DDR4_T_REF_RANGE': 'Normal (0-85)',
'PSU__DDRC__DDR3_T_REF_RANGE': 'NA',
'PSU__DDRC__DDR3L_T_REF_RANGE': 'NA',
'PSU__DDRC__LPDDR3_T_REF_RANGE': 'NA',
'PSU__DDRC__LPDDR4_T_REF_RANGE': 'NA',
'PSU__DDRC__PHY_DBI_MODE': '0',
'PSU__DDRC__DM_DBI': 'DM_NO_DBI',
'PSU__DDRC__COMPONENTS': 'Components',
'PSU__DDRC__PARITY_ENABLE': '0',
'PSU__DDRC__DDR4_CAL_MODE_ENABLE': '0',
'PSU__DDRC__DDR4_CRC_CONTROL': '0',
'PSU__DDRC__FGRM': '1X',
'PSU__DDRC__VENDOR_PART': 'OTHERS',
'PSU__DDRC__SB_TARGET': '16-16-16',
'PSU__DDRC__LP_ASR': 'manual normal',
'PSU__DDRC__DDR4_ADDR_MAPPING': '0',
'PSU__DDRC__SELF_REF_ABORT': '0',
'PSU__DDRC__DERATE_INT_D': '<Select>',
'PSU__DDRC__ADDR_MIRROR': '0',
'PSU__DDRC__EN_2ND_CLK': '0',
'PSU__DDRC__LPDDR3_DUALRANK_SDP': '0',
'PSU__DDRC__PER_BANK_REFRESH': '0',
'PSU__DDRC__ENABLE_DP_SWITCH': '0',
'PSU__DDRC__ENABLE_LP4_SLOWBOOT': '0',
'PSU__DDRC__ENABLE_LP4_HAS_ECC_COMP': '0',
'PSU__DDRC__ENABLE_2T_TIMING': '0',
'PSU__DDRC__RD_DQS_CENTER': '0',
'PSU__DDRC__DQMAP_0_3': '0',
'PSU__DDRC__DQMAP_4_7': '0',
'PSU__DDRC__DQMAP_8_11': '0',
'PSU__DDRC__DQMAP_12_15': '0',
'PSU__DDRC__DQMAP_16_19': '0',
'PSU__DDRC__DQMAP_20_23': '0',
'PSU__DDRC__DQMAP_24_27': '0',
'PSU__DDRC__DQMAP_28_31': '0',
'PSU__DDRC__DQMAP_32_35': '0',
'PSU__DDRC__DQMAP_36_39': '0',
'PSU__DDRC__DQMAP_40_43': '0',
'PSU__DDRC__DQMAP_44_47': '0',
'PSU__DDRC__DQMAP_48_51': '0',
'PSU__DDRC__DQMAP_52_55': '0',
'PSU__DDRC__DQMAP_56_59': '0',
'PSU__DDRC__DQMAP_60_63': '0',
'PSU__DDRC__DQMAP_64_67': '0',
'PSU__DDRC__DQMAP_68_71': '0',
```

```
'PSU__DDR_RAM_HIGHADDR': '0xFFFFFFFF',
'PSU__DDR_RAM_HIGHADDR_OFFSET': '0x800000000',
'PSU__DDR_RAM_LOWADDR_OFFSET': '0x80000000',
'PSU__DDR_QOS_ENABLE': '0',
'PSU__DDR_QOS_PORT0_TYPE': '<Select>',
'PSU__DDR_QOS_PORT1_VN1_TYPE': '<Select>',
'PSU__DDR_QOS_PORT1_VN2_TYPE': '<Select>',
'PSU__DDR_QOS_PORT2_VN1_TYPE': '<Select>',
'PSU__DDR_QOS_PORT2_VN2_TYPE': '<Select>',
'PSU__DDR_QOS_PORT3_TYPE': '<Select>',
'PSU__DDR_QOS_PORT4_TYPE': '<Select>',
'PSU__DDR_QOS_PORT5_TYPE': '<Select>',
'PSU__DDR_QOS_RD_LPR_THRSHLD': None,
'PSU__DDR_QOS_RD_HPR_THRSHLD': None,
'PSU__DDR_QOS_WR_THRSHLD': None,
'PSU__DDR_QOS_HP0_RDQOS': None,
'PSU__DDR_QOS_HP0_WRQOS': None,
'PSU__DDR_QOS_HP1_RDQOS': None,
'PSU__DDR_QOS_HP1_WRQOS': None,
'PSU__DDR_QOS_HP2_RDQOS': None,
'PSU__DDR_QOS_HP2_WRQOS': None,
'PSU__DDR_QOS_HP3_RDQOS': None,
'PSU__DDR_QOS_HP3_WRQOS': None,
'PSU__DDR_QOS_FIX_HP0_RDQOS': None,
'PSU__DDR_QOS_FIX_HP0_WRQOS': None,
'PSU__DDR_QOS_FIX_HP1_RDQOS': None,
'PSU__DDR_QOS_FIX_HP1_WRQOS': None,
'PSU__DDR_QOS_FIX_HP2_RDQOS': None,
'PSU__DDR_QOS_FIX_HP2_WRQOS': None,
'PSU__DDR_QOS_FIX_HP3_RDQOS': None,
'PSU__DDR_QOS_FIX_HP3_WRQOS': None,
'PSU__OVERRIDE_HPX_QOS': '0',
'PSU__FP__POWER__ON': '1',
'PSU__PL__POWER__ON': '1',
'PSU__OCM_BANK0__POWER__ON': '1',
'PSU__OCM_BANK1__POWER__ON': '1',
'PSU__OCM_BANK2__POWER__ON': '1',
'PSU__OCM_BANK3__POWER__ON': '1',
'PSU__TCM0A__POWER__ON': '1',
'PSU__TCM0B__POWER__ON': '1',
'PSU__TCM1A__POWER__ON': '1',
'PSU__TCM1B__POWER__ON': '1',
'PSU__RPU__POWER__ON': '1',
'PSU__L2_BANK0__POWER__ON': '1',
'PSU__GPU_PP0__POWER__ON': '1',
'PSU__GPU_PP1__POWER__ON': '1',
'PSU__ACPU0__POWER__ON': '1',
'PSU__ACPU1__POWER__ON': '1',
'PSU__ACPU2__POWER__ON': '1',
'PSU__ACPU3__POWER__ON': '1',
'PSU__UART0__PERIPHERAL__ENABLE': '0',
'PSU__UART0__PERIPHERAL__IO': '<Select>',
'PSU__UART0__MODEM__ENABLE': '0',
'PSU__UART1__PERIPHERAL__ENABLE': '0',
'PSU__UART1__PERIPHERAL__IO': '<Select>',
'PSU__UART1__MODEM__ENABLE': '0',
'PSU__UART0_LOOP_UART1__ENABLE': '0',
'PSU__USB0__PERIPHERAL__ENABLE': '0',
'PSU__USB0__PERIPHERAL__IO': '<Select>',
'PSU__USB0__RESET__ENABLE': '0',
'PSU__USB0__RESET__IO': '<Select>',
'PSU__USB__RESET__MODE': '<Select>',
'PSU__USB__RESET__POLARITY': '<Select>',
'PSU__USB1__PERIPHERAL__ENABLE': '0',
'PSU__USB1__PERIPHERAL__IO': '<Select>',
'PSU__USB1__RESET__ENABLE': '0',
'PSU__USB1__RESET__IO': '<Select>',
'PSU__USB3_0__PERIPHERAL__ENABLE': '0',
'PSU__USB3_0__PERIPHERAL__IO': '<Select>',
'PSU__USB3_1__PERIPHERAL__ENABLE': '0',
'PSU__USB3_1__PERIPHERAL__IO': '<Select>',
'PSU__USB3_0__EMIO__ENABLE': '0',
'PSU__USB2_0__EMIO__ENABLE': '0',
'PSU__USB3_1__EMIO__ENABLE': '0',
'PSU__USB2_1__EMIO__ENABLE': '0',
'PSU__USE__USB3_0_HUB': '0',
'PSU__USE__USB3_1_HUB': '0',
'PSU__USE__ADMA': '0',
'PSU__USE__M_AXI_GP0': '1',
'PSU__M_AXI_GP0_SUPPORTS_NARROW_BURST': '1',
'PSU__MAXIGP0__DATA_WIDTH': '128',
'PSU__USE__M_AXI_GP1': '1',
'PSU__M_AXI_GP1_SUPPORTS_NARROW_BURST': '1',
'PSU__MAXIGP1__DATA_WIDTH': '128',
'PSU__USE__M_AXI_GP2': '0',
'PSU__M_AXI_GP2_SUPPORTS_NARROW_BURST': '1',
'PSU__MAXIGP2__DATA_WIDTH': '64',
'PSU__USE__S_AXI_ACP': '0',
'PSU__USE__S_AXI_GP0': '0',
'PSU__USE_DIFF_RW_CLK_GP0': '0',
'PSU__SAXIGP0__DATA_WIDTH': '128',
'PSU__USE__S_AXI_GP1': '0',
'PSU__USE_DIFF_RW_CLK_GP1': '0',
```

```
                'PSU__SAXIGP1__DATA_WIDTH': '128',
                'PSU__USE__S_AXI_GP2': '1',
                'PSU__USE_DIFF_RW_CLK_GP2': '0',
                'PSU__SAXIGP2__DATA_WIDTH': '64',
                'PSU__USE__S_AXI_GP3': '0',
                'PSU__USE_DIFF_RW_CLK_GP3': '0',
                'PSU__SAXIGP3__DATA_WIDTH': '128',
                'PSU__USE__S_AXI_GP4': '1',
                'PSU__USE_DIFF_RW_CLK_GP4': '0',
                'PSU__SAXIGP4__DATA_WIDTH': '64',
                'PSU__USE__S_AXI_GP5': '0',
                'PSU__USE_DIFF_RW_CLK_GP5': '0',
                'PSU__SAXIGP5__DATA_WIDTH': '128',
                'PSU__USE__S_AXI_GP6': '0',
                'PSU__USE_DIFF_RW_CLK_GP6': '0',
                'PSU__SAXIGP6__DATA_WIDTH': '128',
                'PSU__USE__S_AXI_ACE': '0',
                'PSU__TRACE_PIPELINE_WIDTH': '8',
                'PSU__EN_EMIO_TRACE': '0',
                'PSU__USE__AUDIO': '0',
                'PSU__USE__VIDEO': '0',
                'PSU__USE__PROC_EVENT_BUS': '0',
                'PSU__USE__FTM': '0',
                'PSU__USE__CROSS_TRIGGER': '0',
                'PSU__FTM__CTI_IN_0': '0',
                'PSU__FTM__CTI_IN_1': '0',
                'PSU__FTM__CTI_IN_2': '0',
                'PSU__FTM__CTI_IN_3': '0',
                'PSU__FTM__CTI_OUT_0': '0',
                'PSU__FTM__CTI_OUT_1': '0',
                'PSU__FTM__CTI_OUT_2': '0',
                'PSU__FTM__CTI_OUT_3': '0',
                'PSU__FTM__GPO': '0',
                'PSU__FTM__GPI': '0',
                'PSU__USE__GDMA': '0',
                'PSU__USE__IRQ': '0',
                'PSU__USE__IRQ0': '1',
                'PSU__USE__IRQ1': '0',
                'PSU__USE__CLK0': '0',
                'PSU__USE__CLK1': '0',
                'PSU__USE__CLK2': '0',
                'PSU__USE__CLK3': '0',
                'PSU__USE__RST0': '0',
                'PSU__USE__RST1': '0',
                'PSU__USE__RST2': '0',
                'PSU__USE__RST3': '0',
                'PSU__USE__FABRIC__RST': '1',
                'PSU__USE__RTC': '0',
                'PSU__PRESET_APPLIED': '1',
                'PSU__USE__EVENT_RPU': '0',
                'PSU__USE__APU_LEGACY_INTERRUPT': '0',
                'PSU__USE__RPU_LEGACY_INTERRUPT': '0',
                'PSU__USE__STM': '0',
                'PSU__USE__DEBUG__TEST': '0',
                'PSU__HIGH_ADDRESS__ENABLE': '1',
                'PSU__DDR_HIGH_ADDRESS_GUI_ENABLE': '1',
                'PSU__EXPAND__LOWER_LPS_SLAVES': '0',
                'PSU__EXPAND__CORESIGHT': '0',
                'PSU__EXPAND__GIC': '0',
                'PSU__EXPAND__FPD_SLAVES': '0',
                'PSU__EXPAND__UPPER_LPS_SLAVES': '0',
                'PSU_MIO_0_PULLUPDOWN': 'pullup',
                'PSU_MIO_0_DRIVE_STRENGTH': '4',
                'PSU_MIO_0_POLARITY': 'Default',
                'PSU_MIO_0_INPUT_TYPE': 'cmos',
                'PSU_MIO_0_SLEW': 'slow',
                'PSU_MIO_0_DIRECTION': 'out',
                'PSU_MIO_1_PULLUPDOWN': 'pullup',
                'PSU_MIO_1_DRIVE_STRENGTH': '4',
                'PSU_MIO_1_POLARITY': 'Default',
                'PSU_MIO_1_INPUT_TYPE': 'cmos',
                'PSU_MIO_1_SLEW': 'slow',
                'PSU_MIO_1_DIRECTION': 'inout',
                'PSU_MIO_2_PULLUPDOWN': 'pullup',
                'PSU_MIO_2_DRIVE_STRENGTH': '4',
                'PSU_MIO_2_POLARITY': 'Default',
                'PSU_MIO_2_INPUT_TYPE': 'cmos',
                'PSU_MIO_2_SLEW': 'slow',
                'PSU_MIO_2_DIRECTION': 'inout',
                'PSU_MIO_3_PULLUPDOWN': 'pullup',
                'PSU_MIO_3_DRIVE_STRENGTH': '4',
                'PSU_MIO_3_POLARITY': 'Default',
                'PSU_MIO_3_INPUT_TYPE': 'cmos',
                'PSU_MIO_3_SLEW': 'slow',
                'PSU_MIO_3_DIRECTION': 'inout',
                'PSU_MIO_4_PULLUPDOWN': 'pullup',
                'PSU_MIO_4_DRIVE_STRENGTH': '4',
                'PSU_MIO_4_POLARITY': 'Default',
                'PSU_MIO_4_INPUT_TYPE': 'cmos',
                'PSU_MIO_4_SLEW': 'slow',
                'PSU_MIO_4_DIRECTION': 'inout',
                'PSU_MIO_5_PULLUPDOWN': 'pullup',
                'PSU_MIO_5_DRIVE_STRENGTH': '4',
```

```
                'PSU_MIO_5_POLARITY': 'Default',
                'PSU_MIO_5_INPUT_TYPE': 'cmos',
                'PSU_MIO_5_SLEW': 'slow',
                'PSU_MIO_5_DIRECTION': 'out',
                'PSU_MIO_6_PULLUPDOWN': 'pullup',
                'PSU_MIO_6_DRIVE_STRENGTH': '4',
                'PSU_MIO_6_POLARITY': 'Default',
                'PSU_MIO_6_INPUT_TYPE': 'cmos',
                'PSU_MIO_6_SLEW': 'slow',
                'PSU_MIO_6_DIRECTION': 'inout',
                'PSU_MIO_7_PULLUPDOWN': 'pullup',
                'PSU_MIO_7_DRIVE_STRENGTH': '4',
                'PSU_MIO_7_POLARITY': 'Default',
                'PSU_MIO_7_INPUT_TYPE': 'cmos',
                'PSU_MIO_7_SLEW': 'slow',
                'PSU_MIO_7_DIRECTION': 'inout',
                'PSU_MIO_8_PULLUPDOWN': 'pullup',
                'PSU_MIO_8_DRIVE_STRENGTH': '4',
                'PSU_MIO_8_POLARITY': 'Default',
                'PSU_MIO_8_INPUT_TYPE': 'cmos',
                'PSU_MIO_8_SLEW': 'slow',
                'PSU_MIO_8_DIRECTION': 'inout',
                'PSU_MIO_9_PULLUPDOWN': 'pullup',
                'PSU_MIO_9_DRIVE_STRENGTH': '4',
                'PSU_MIO_9_POLARITY': 'Default',
                'PSU_MIO_9_INPUT_TYPE': 'cmos',
                'PSU_MIO_9_SLEW': 'slow',
                'PSU_MIO_9_DIRECTION': 'inout',
                'PSU_MIO_10_PULLUPDOWN': 'pullup',
                'PSU_MIO_10_DRIVE_STRENGTH': '4',
                'PSU_MIO_10_POLARITY': 'Default',
                'PSU_MIO_10_INPUT_TYPE': 'cmos',
                'PSU_MIO_10_SLEW': 'slow',
                'PSU_MIO_10_DIRECTION': 'inout',
                'PSU_MIO_11_PULLUPDOWN': 'pullup',
                'PSU_MIO_11_DRIVE_STRENGTH': '4',
                'PSU_MIO_11_POLARITY': 'Default',
                'PSU_MIO_11_INPUT_TYPE': 'cmos',
                'PSU_MIO_11_SLEW': 'slow',
                'PSU_MIO_11_DIRECTION': 'inout',
                'PSU_MIO_12_PULLUPDOWN': 'pullup',
                'PSU_MIO_12_DRIVE_STRENGTH': '4',
                'PSU_MIO_12_POLARITY': 'Default',
                'PSU_MIO_12_INPUT_TYPE': 'cmos',
                'PSU_MIO_12_SLEW': 'slow',
                'PSU_MIO_12_DIRECTION': 'inout',
                'PSU_MIO_13_PULLUPDOWN': 'pullup',
                'PSU_MIO_13_DRIVE_STRENGTH': '4',
                'PSU_MIO_13_POLARITY': 'Default',
                'PSU_MIO_13_INPUT_TYPE': 'cmos',
                'PSU_MIO_13_SLEW': 'slow',
                'PSU_MIO_13_DIRECTION': 'inout',
                'PSU_MIO_14_PULLUPDOWN': 'pullup',
                'PSU_MIO_14_DRIVE_STRENGTH': '4',
                'PSU_MIO_14_POLARITY': 'Default',
                'PSU_MIO_14_INPUT_TYPE': 'cmos',
                'PSU_MIO_14_SLEW': 'slow',
                'PSU_MIO_14_DIRECTION': 'inout',
                'PSU_MIO_15_PULLUPDOWN': 'pullup',
                'PSU_MIO_15_DRIVE_STRENGTH': '4',
                'PSU_MIO_15_POLARITY': 'Default',
                'PSU_MIO_15_INPUT_TYPE': 'cmos',
                'PSU_MIO_15_SLEW': 'slow',
                'PSU_MIO_15_DIRECTION': 'inout',
                'PSU_MIO_16_PULLUPDOWN': 'pullup',
                'PSU_MIO_16_DRIVE_STRENGTH': '4',
                'PSU_MIO_16_POLARITY': 'Default',
                'PSU_MIO_16_INPUT_TYPE': 'cmos',
                'PSU_MIO_16_SLEW': 'slow',
                'PSU_MIO_16_DIRECTION': 'inout',
                'PSU_MIO_17_PULLUPDOWN': 'pullup',
                'PSU_MIO_17_DRIVE_STRENGTH': '4',
                'PSU_MIO_17_POLARITY': 'Default',
                'PSU_MIO_17_INPUT_TYPE': 'cmos',
                'PSU_MIO_17_SLEW': 'slow',
                'PSU_MIO_17_DIRECTION': 'inout',
                'PSU_MIO_18_PULLUPDOWN': 'pullup',
                'PSU_MIO_18_DRIVE_STRENGTH': '4',
                'PSU_MIO_18_POLARITY': 'Default',
                'PSU_MIO_18_INPUT_TYPE': 'cmos',
                'PSU_MIO_18_SLEW': 'slow',
                'PSU_MIO_18_DIRECTION': 'inout',
                'PSU_MIO_19_PULLUPDOWN': 'pullup',
                'PSU_MIO_19_DRIVE_STRENGTH': '4',
                'PSU_MIO_19_POLARITY': 'Default',
                'PSU_MIO_19_INPUT_TYPE': 'cmos',
                'PSU_MIO_19_SLEW': 'slow',
                'PSU_MIO_19_DIRECTION': 'inout',
                'PSU_MIO_20_PULLUPDOWN': 'pullup',
                'PSU_MIO_20_DRIVE_STRENGTH': '4',
                'PSU_MIO_20_POLARITY': 'Default',
                'PSU_MIO_20_INPUT_TYPE': 'cmos',
                'PSU_MIO_20_SLEW': 'slow',
```

```
'PSU_MIO_20_DIRECTION': 'inout',
'PSU_MIO_21_PULLUPDOWN': 'pullup',
'PSU_MIO_21_DRIVE_STRENGTH': '4',
'PSU_MIO_21_POLARITY': 'Default',
'PSU_MIO_21_INPUT_TYPE': 'cmos',
'PSU_MIO_21_SLEW': 'slow',
'PSU_MIO_21_DIRECTION': 'inout',
'PSU_MIO_22_PULLUPDOWN': 'pullup',
'PSU_MIO_22_DRIVE_STRENGTH': '4',
'PSU_MIO_22_POLARITY': 'Default',
'PSU_MIO_22_INPUT_TYPE': 'cmos',
'PSU_MIO_22_SLEW': 'slow',
'PSU_MIO_22_DIRECTION': 'inout',
'PSU_MIO_23_PULLUPDOWN': 'pullup',
'PSU_MIO_23_DRIVE_STRENGTH': '4',
'PSU_MIO_23_POLARITY': 'Default',
'PSU_MIO_23_INPUT_TYPE': 'cmos',
'PSU_MIO_23_SLEW': 'slow',
'PSU_MIO_23_DIRECTION': 'inout',
'PSU_MIO_24_PULLUPDOWN': 'pullup',
'PSU_MIO_24_DRIVE_STRENGTH': '4',
'PSU_MIO_24_POLARITY': 'Default',
'PSU_MIO_24_INPUT_TYPE': 'cmos',
'PSU_MIO_24_SLEW': 'slow',
'PSU_MIO_24_DIRECTION': 'inout',
'PSU_MIO_25_PULLUPDOWN': 'pullup',
'PSU_MIO_25_DRIVE_STRENGTH': '4',
'PSU_MIO_25_POLARITY': 'Default',
'PSU_MIO_25_INPUT_TYPE': 'cmos',
'PSU_MIO_25_SLEW': 'slow',
'PSU_MIO_25_DIRECTION': 'inout',
'PSU_MIO_26_PULLUPDOWN': 'pullup',
'PSU_MIO_26_DRIVE_STRENGTH': '12',
'PSU_MIO_26_POLARITY': 'Default',
'PSU_MIO_26_INPUT_TYPE': 'cmos',
'PSU_MIO_26_SLEW': 'fast',
'PSU_MIO_26_DIRECTION': 'in',
'PSU_MIO_27_PULLUPDOWN': 'pullup',
'PSU_MIO_27_DRIVE_STRENGTH': '4',
'PSU_MIO_27_POLARITY': 'Default',
'PSU_MIO_27_INPUT_TYPE': 'cmos',
'PSU_MIO_27_SLEW': 'slow',
'PSU_MIO_27_DIRECTION': 'inout',
'PSU_MIO_28_PULLUPDOWN': 'pullup',
'PSU_MIO_28_DRIVE_STRENGTH': '12',
'PSU_MIO_28_POLARITY': 'Default',
'PSU_MIO_28_INPUT_TYPE': 'cmos',
'PSU_MIO_28_SLEW': 'fast',
'PSU_MIO_28_DIRECTION': 'inout',
'PSU_MIO_29_PULLUPDOWN': 'pullup',
'PSU_MIO_29_DRIVE_STRENGTH': '4',
'PSU_MIO_29_POLARITY': 'Default',
'PSU_MIO_29_INPUT_TYPE': 'cmos',
'PSU_MIO_29_SLEW': 'slow',
'PSU_MIO_29_DIRECTION': 'inout',
'PSU_MIO_30_PULLUPDOWN': 'pullup',
'PSU_MIO_30_DRIVE_STRENGTH': '12',
'PSU_MIO_30_POLARITY': 'Default',
'PSU_MIO_30_INPUT_TYPE': 'cmos',
'PSU_MIO_30_SLEW': 'fast',
'PSU_MIO_30_DIRECTION': 'inout',
'PSU_MIO_31_PULLUPDOWN': 'pullup',
'PSU_MIO_31_DRIVE_STRENGTH': '12',
'PSU_MIO_31_POLARITY': 'Default',
'PSU_MIO_31_INPUT_TYPE': 'cmos',
'PSU_MIO_31_SLEW': 'fast',
'PSU_MIO_31_DIRECTION': 'in',
'PSU_MIO_32_PULLUPDOWN': 'pullup',
'PSU_MIO_32_DRIVE_STRENGTH': '4',
'PSU_MIO_32_POLARITY': 'Default',
'PSU_MIO_32_INPUT_TYPE': 'cmos',
'PSU_MIO_32_SLEW': 'slow',
'PSU_MIO_32_DIRECTION': 'out',
'PSU_MIO_33_PULLUPDOWN': 'pullup',
'PSU_MIO_33_DRIVE_STRENGTH': '4',
'PSU_MIO_33_POLARITY': 'Default',
'PSU_MIO_33_INPUT_TYPE': 'cmos',
'PSU_MIO_33_SLEW': 'slow',
'PSU_MIO_33_DIRECTION': 'out',
'PSU_MIO_34_PULLUPDOWN': 'pullup',
'PSU_MIO_34_DRIVE_STRENGTH': '4',
'PSU_MIO_34_POLARITY': 'Default',
'PSU_MIO_34_INPUT_TYPE': 'cmos',
'PSU_MIO_34_SLEW': 'slow',
'PSU_MIO_34_DIRECTION': 'out',
'PSU_MIO_35_PULLUPDOWN': 'pullup',
'PSU_MIO_35_DRIVE_STRENGTH': '4',
'PSU_MIO_35_POLARITY': 'Default',
'PSU_MIO_35_INPUT_TYPE': 'cmos',
'PSU_MIO_35_SLEW': 'slow',
'PSU_MIO_35_DIRECTION': 'out',
'PSU_MIO_36_PULLUPDOWN': 'pullup',
'PSU_MIO_36_DRIVE_STRENGTH': '4',
```

```
                'PSU_MIO_36_POLARITY': 'Default',
                'PSU_MIO_36_INPUT_TYPE': 'cmos',
                'PSU_MIO_36_SLEW': 'slow',
                'PSU_MIO_36_DIRECTION': 'inout',
                'PSU_MIO_37_PULLUPDOWN': 'pullup',
                'PSU_MIO_37_DRIVE_STRENGTH': '12',
                'PSU_MIO_37_POLARITY': 'Default',
                'PSU_MIO_37_INPUT_TYPE': 'cmos',
                'PSU_MIO_37_SLEW': 'fast',
                'PSU_MIO_37_DIRECTION': 'inout',
                'PSU_MIO_38_PULLUPDOWN': 'pullup',
                'PSU_MIO_38_DRIVE_STRENGTH': '4',
                'PSU_MIO_38_POLARITY': 'Default',
                'PSU_MIO_38_INPUT_TYPE': 'cmos',
                'PSU_MIO_38_SLEW': 'slow',
                'PSU_MIO_38_DIRECTION': 'inout',
                'PSU_MIO_39_PULLUPDOWN': 'pullup',
                'PSU_MIO_39_DRIVE_STRENGTH': '4',
                'PSU_MIO_39_POLARITY': 'Default',
                'PSU_MIO_39_INPUT_TYPE': 'cmos',
                'PSU_MIO_39_SLEW': 'slow',
                'PSU_MIO_39_DIRECTION': 'inout',
                'PSU_MIO_40_PULLUPDOWN': 'pullup',
                'PSU_MIO_40_DRIVE_STRENGTH': '4',
                'PSU_MIO_40_POLARITY': 'Default',
                'PSU_MIO_40_INPUT_TYPE': 'cmos',
                'PSU_MIO_40_SLEW': 'slow',
                'PSU_MIO_40_DIRECTION': 'inout',
                'PSU_MIO_41_PULLUPDOWN': 'pullup',
                'PSU_MIO_41_DRIVE_STRENGTH': '4',
                'PSU_MIO_41_POLARITY': 'Default',
                'PSU_MIO_41_INPUT_TYPE': 'cmos',
                'PSU_MIO_41_SLEW': 'slow',
                'PSU_MIO_41_DIRECTION': 'inout',
                'PSU_MIO_42_PULLUPDOWN': 'pullup',
                'PSU_MIO_42_DRIVE_STRENGTH': '4',
                'PSU_MIO_42_POLARITY': 'Default',
                'PSU_MIO_42_INPUT_TYPE': 'cmos',
                'PSU_MIO_42_SLEW': 'slow',
                'PSU_MIO_42_DIRECTION': 'inout',
                'PSU_MIO_43_PULLUPDOWN': 'pullup',
                'PSU_MIO_43_DRIVE_STRENGTH': '4',
                'PSU_MIO_43_POLARITY': 'Default',
                'PSU_MIO_43_INPUT_TYPE': 'cmos',
                'PSU_MIO_43_SLEW': 'slow',
                'PSU_MIO_43_DIRECTION': 'inout',
                'PSU_MIO_44_PULLUPDOWN': 'pullup',
                'PSU_MIO_44_DRIVE_STRENGTH': '4',
                'PSU_MIO_44_POLARITY': 'Default',
                'PSU_MIO_44_INPUT_TYPE': 'cmos',
                'PSU_MIO_44_SLEW': 'slow',
                'PSU_MIO_44_DIRECTION': 'inout',
                'PSU_MIO_45_PULLUPDOWN': 'pullup',
                'PSU_MIO_45_DRIVE_STRENGTH': '12',
                'PSU_MIO_45_POLARITY': 'Default',
                'PSU_MIO_45_INPUT_TYPE': 'cmos',
                'PSU_MIO_45_SLEW': 'fast',
                'PSU_MIO_45_DIRECTION': 'inout',
                'PSU_MIO_46_PULLUPDOWN': 'pullup',
                'PSU_MIO_46_DRIVE_STRENGTH': '4',
                'PSU_MIO_46_POLARITY': 'Default',
                'PSU_MIO_46_INPUT_TYPE': 'cmos',
                'PSU_MIO_46_SLEW': 'slow',
                'PSU_MIO_46_DIRECTION': 'inout',
                'PSU_MIO_47_PULLUPDOWN': 'pullup',
                'PSU_MIO_47_DRIVE_STRENGTH': '4',
                'PSU_MIO_47_POLARITY': 'Default',
                'PSU_MIO_47_INPUT_TYPE': 'cmos',
                'PSU_MIO_47_SLEW': 'slow',
                'PSU_MIO_47_DIRECTION': 'inout',
                'PSU_MIO_48_PULLUPDOWN': 'pullup',
                'PSU_MIO_48_DRIVE_STRENGTH': '4',
                'PSU_MIO_48_POLARITY': 'Default',
                'PSU_MIO_48_INPUT_TYPE': 'cmos',
                'PSU_MIO_48_SLEW': 'slow',
                'PSU_MIO_48_DIRECTION': 'inout',
                'PSU_MIO_49_PULLUPDOWN': 'pullup',
                'PSU_MIO_49_DRIVE_STRENGTH': '4',
                'PSU_MIO_49_POLARITY': 'Default',
                'PSU_MIO_49_INPUT_TYPE': 'cmos',
                'PSU_MIO_49_SLEW': 'slow',
                'PSU_MIO_49_DIRECTION': 'inout',
                'PSU_MIO_50_PULLUPDOWN': 'pullup',
                'PSU_MIO_50_DRIVE_STRENGTH': '4',
                'PSU_MIO_50_POLARITY': 'Default',
                'PSU_MIO_50_INPUT_TYPE': 'cmos',
                'PSU_MIO_50_SLEW': 'slow',
                'PSU_MIO_50_DIRECTION': 'inout',
                'PSU_MIO_51_PULLUPDOWN': 'pullup',
                'PSU_MIO_51_DRIVE_STRENGTH': '4',
                'PSU_MIO_51_POLARITY': 'Default',
                'PSU_MIO_51_INPUT_TYPE': 'cmos',
                'PSU_MIO_51_SLEW': 'slow',
```

```
 'PSU_MIO_51_DIRECTION': 'inout',
 'PSU_MIO_52_PULLUPDOWN': 'pullup',
 'PSU_MIO_52_DRIVE_STRENGTH': '12',
 'PSU_MIO_52_POLARITY': 'Default',
 'PSU_MIO_52_INPUT_TYPE': 'cmos',
 'PSU_MIO_52_SLEW': 'fast',
 'PSU_MIO_52_DIRECTION': '<Select>',
 'PSU_MIO_53_PULLUPDOWN': 'pullup',
 'PSU_MIO_53_DRIVE_STRENGTH': '12',
 'PSU_MIO_53_POLARITY': 'Default',
 'PSU_MIO_53_INPUT_TYPE': 'cmos',
 'PSU_MIO_53_SLEW': 'fast',
 'PSU_MIO_53_DIRECTION': '<Select>',
 'PSU_MIO_54_PULLUPDOWN': 'pullup',
 'PSU_MIO_54_DRIVE_STRENGTH': '4',
 'PSU_MIO_54_POLARITY': 'Default',
 'PSU_MIO_54_INPUT_TYPE': 'cmos',
 'PSU_MIO_54_SLEW': 'slow',
 'PSU_MIO_54_DIRECTION': '<Select>',
 'PSU_MIO_55_PULLUPDOWN': 'pullup',
 'PSU_MIO_55_DRIVE_STRENGTH': '12',
 'PSU_MIO_55_POLARITY': 'Default',
 'PSU_MIO_55_INPUT_TYPE': 'cmos',
 'PSU_MIO_55_SLEW': 'fast',
 'PSU_MIO_55_DIRECTION': '<Select>',
 ...},
'driver': pynq.overlay.DefaultIP,
'device': <pynq.pl_server.embedded_device.EmbeddedDevice at 0xffffb39cfa60>}}
```

The HLS has the default name from the Vivado project of *example_0*.

Check *help* for the HLS IP:

In [21]:
```
ol.example0_0?
```

```
Type:        DefaultIP
String form: <pynq.overlay.DefaultIP object at 0xffff768b6560>
File:        /usr/local/share/pynq-venv/lib/python3.10/site-packages/pynq/overlay.py
Docstring:
Driver for an IP without a more specific driver

This driver wraps an MMIO device and provides a base class
for more specific drivers written later. It also provides
access to GPIO outputs and interrupts inputs via attributes. More specific
drivers should inherit from `DefaultIP` and include a
`bindto` entry containing all of the IP that the driver
should bind to. Subclasses meeting these requirements will
automatically be registered.

Attributes
----------
mmio : pynq.MMIO
    Underlying MMIO driver for the device
_interrupts : dict
    Subset of the PL.interrupt_pins related to this IP
_gpio : dict
    Subset of the PL.gpio_dict related to this IP
```

This tells us that this is not a known IP (type is **DefaultIP**) and will get assigned a default driver in PYNQ. The default driver provides MMIO read/write capability.

Create aliases

Using the labels for the HLS IP and DMA listed above, we can create aliases which will make it easier to write and read the rest of the code in this example.

In [22]:
```
dma = ol.axi_dma_0
dma_send = ol.axi_dma_0.sendchannel
dma_recv = ol.axi_dma_0.recvchannel

hls_ip = ol.example0_0
```

Check the status of the HLS IP

In [23]:
```
hls_ip.register_map
```

Out[23]:
```
RegisterMap {
  CTRL = Register(AP_START=0, AP_DONE=0, AP_IDLE=1, AP_READY=0, RESERVED_1=0, AUTO_RESTART=0, RESERVED_2=0, INTERRUPT
=0, RESERVED_3=0),
  GIER = Register(Enable=0, RESERVED=0),
  IP_IER = Register(CHAN0_INT_EN=0, CHAN1_INT_EN=0, RESERVED_0=0),
  IP_ISR = Register(CHAN0_INT_ST=0, CHAN1_INT_ST=0, RESERVED_0=0)
}
```

Note that the HLS IP is not started yet (AP_START=0). You can also see the IP is *idle* (AP_IDLE=1).

We will start the HLS IP and then start some transfers from the DMA.

We could initiate the DMA transfers first if we preferred. The DMA transfers would *stall* until the IP is started.

Start the HLS IP

We can start the HLS IP by writing 0x81 to the control register. This will set bit 0 (AP_START) to "1" and bit 7 (AUTO_RESTART) to "1". AUTO_RESTART means the IP will run continuously. If we don't set this then after the IP completes one full operation or iteration, it will stop and wait until AP_START is set again. We would have to set this every time we want the IP to process some data.

In [24]:
```
CONTROL_REGISTER = 0x0
hls_ip.write(CONTROL_REGISTER, 0x81) # 0x81 will set bit 0
```

Check the correct bits have been set.

In [25]:
```
hls_ip.register_map
```

Out[25]:
```
RegisterMap {
  CTRL = Register(AP_START=1, AP_DONE=0, AP_IDLE=0, AP_READY=0, RESERVED_1=0, AUTO_RESTART=1, RESERVED_2=0, INTERRUPT
=0, RESERVED_3=0),
  GIER = Register(Enable=0, RESERVED=0),
  IP_IER = Register(CHAN0_INT_EN=0, CHAN1_INT_EN=0, RESERVED_0=0),
  IP_ISR = Register(CHAN0_INT_ST=0, CHAN1_INT_ST=0, RESERVED_0=0)
}
```

DMA send

Now we will send some data from DRAM to the HLS IP. Once the HLS IP is started, the steps are the same as the previous DMA tutorial.

Note the array used below is uint32. This was selected to match the type of data width of the HLS IP and the widths used for the DMA. The DMA will transfer blocks of data and may "reformat" it based on the internal data widths in the hardware. The array we create in the notebook effects how the data is formatted in Python. This is an area that can cause problems so it is worthwhile checking you understand your data formats and data movement in your hardware. If you see the wrong

In [26]:
```
from pynq import allocate
import numpy as np

data_size = 100
input_buffer = allocate(shape=(data_size,), dtype=np.int_)
```

Initialize the array.

In [27]:
```
for i in range(data_size):
    input_buffer[i] = i+100
```

Start the DMA transfer

In [28]:
```
dma_send.transfer(input_buffer)
```

DMA receive

Readback data from the HLS IP and store in DRAM. Start by creating the output buffer

In [29]:
```
output_buffer = allocate(shape=(data_size,), dtype=np.uint32)
```

In [30]:
```
dma_recv.transfer(output_buffer)
```

Print first few values of buffer

The result from the HLS IP should be "i"+5

In [31]:
```
for i in range(10):
    print('0x' + format(output_buffer[i], '02x'))
```

```
0x69
0x05
0x6a
0x05
0x6b
0x05
0x6c
0x05
0x6d
0x05
```

In [32]:
```
# Send more data to DMA
```

```
for i in range(data_size):
    input_buffer[i] = i+10



dma_send.transfer(input_buffer)
dma_recv.transfer(output_buffer)
for i in range(10):
    print('0x' + format(output_buffer[i], '02x'))
```

```
---------------------------------------------------------------------------
RuntimeError                              Traceback (most recent call last)
Input In [32], in <cell line: 9>()
      4 for i in range(data_size):
      5     input_buffer[i] = i+10
----> 9 dma_send.transfer(input_buffer)
     10 dma_recv.transfer(output_buffer)
     11 for i in range(10):

File /usr/local/share/pynq-venv/lib/python3.10/site-packages/pynq/lib/dma.py:136, in _SDMAChannel.transfer(self, array, start, nbytes)
    107 """Transfer memory with the DMA
    108
    109 Transfer must only be called when the channel is idle.
   (...)
    133
    134 """
    135 if not self.running:
--> 136     raise RuntimeError("DMA channel not started")
    137 if not self.idle and not self._first_transfer:
    138     raise RuntimeError("DMA channel not idle")

RuntimeError: DMA channel not started
```

Verify that the arrays are equal

In [33]:
```
print("Arrays are equal: {}".format(np.array_equal(input_buffer, output_buffer-5)))
```

```
Arrays are equal: False
```

Free all the memory buffers

Don't forget to free the memory buffers to avoid memory leaks!

In [34]:
```
del input_buffer, output_buffer
```

Summary

In this tutorial you saw how to create a HLS IP with AXI Streams, incorporate this IP into a Vivado design and connected the AXI streams to a DMA, and how to use the HLS IP from PYNQ. The HLS design was very simple, incrementing the input value and writing it to the output. You should be able to see how you can create more advanced HLS kernels. You can also have multiple AXI stream interfaces in your HLS kernel by adding more input or output parameters.

The AXI DMA used in this design supports one AXI stream input and/or one AXI output stream. You can use multiple DMAs if you need to send data to additional AXI stream interfaces.

In [ ]: