DMA datasheet

DMA team

Contents

1 Theory of operation

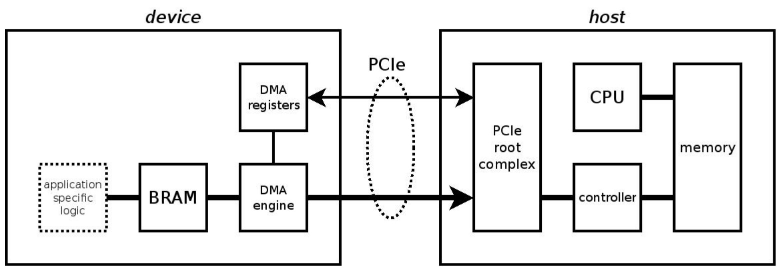

Figure 1: DMA diagram

The Direct Memory Access (DMA) engine transfers an internal device memory contents to a 64 bits host memory over PCIe. A memory transfer runs from start to completion wihtout requiring the host CPU assistance, making it available for other computation tasks.

In this device, the DMA is connected to an internal 32KB byte addressable BRAM memory whose contents are initialized once at main time with an increasing pattern such that:

$$bram[i] = seed + i$$

seed is a user provided value used to randomize the memory contents.

2 Hardware programming interface

The DMA engine PCIe endpoint is mapped at the base address 1, starting at offset 0x0. Only 32 bits aligned accesses are supported.

5 registers (figure 2) are used to interact with the DMA:

- DMA_REG_CTL, controls the engine operations,
- DMA_REG_STA, informs on the engine status,
- DMA_REG_ADx, a pair holding the 64 bits destination address,
- DMA_REG_BAZ, user provided seed to fill internal memory.

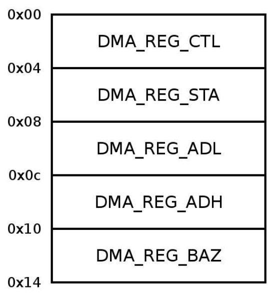

Figure 2: DMA registers

Registers are described in the following sections.

2.1 DMA_REG_CTL

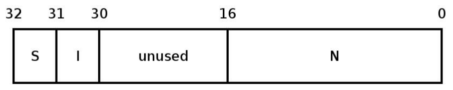

Figure 3: DMA control register

name	*offset (bits)*	*size (bits)*	*description*
S	31	1	set to 1 to start the transfer. self clears.
I	30	1	set to 1 to enable interrupt at end of transfer. default to 0.
N	0	16	size of the transfer, in bytes.

Table 1: DMA_REG_CTL RW register fields

2.2 DMA_REG_STA

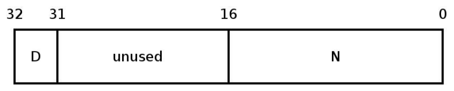

Figure 4: DMA status register

name	*offset (bits)*	*size (bits)*	*description*
D	31	1	automatically set to 1 when a transfer is done. 0 if a transfer is running.
N	0	16	size (in bytes) actually transfered. valid only when DMA not running (ie. R cleared).

Table 2: DMA_REG_STA RO register fields

2.3 DMA_REG_ADx

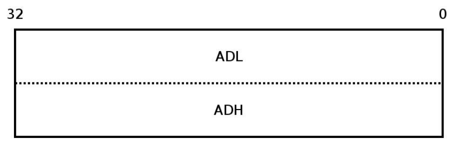

Figure 5: DMA destination address register

name	*offset (bits)*	*size (bits)*	*description*
ADL	0	32	destination address low part (32 least significant bits).
ADH	0	32	destination address high part (32 most significant bits).

Table 3: DMA_REG_ADX RW register fields

2.4 DMA_REG_BAZ

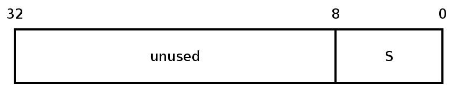

Figure 6: DMA seed register

name	*offset (bits)*	*size (bits)*	*description*
S	0	8	the user provided seed. default to 0.

Table 4: DMA_REG_BAZ RW register fields

3 Application notes

3.1 Programming procedure

When programming the DMA to perform a memory transfer, a user is expected to follow this procedure:

1. the user eventually sets a seed in the DMA_REG_BAZ S field,
2. the user sets DMA_REG_ADx with the 64 bits destination address,
3. the user sets the DMA_REG_CTL N field with the byte count to transfer,
4. the user eventually sets the DMA_REG_CTL I bit if an interrupt is required at the end of the transfer,
5. the user sets the DMA_REG_CTL S bit to start the transfer,
6. the DMA set the DMA_REG_STA D bit upon transfer completion. An interrupt is generated as indicated by the user,
7. the user reads the byte count actually transfered in the DMA_REG_STA N field.