Microprocessors & Microcontrollers
: Arm Cortex M0+
(Using RP2040)

ESM_23

Introduction to DMA

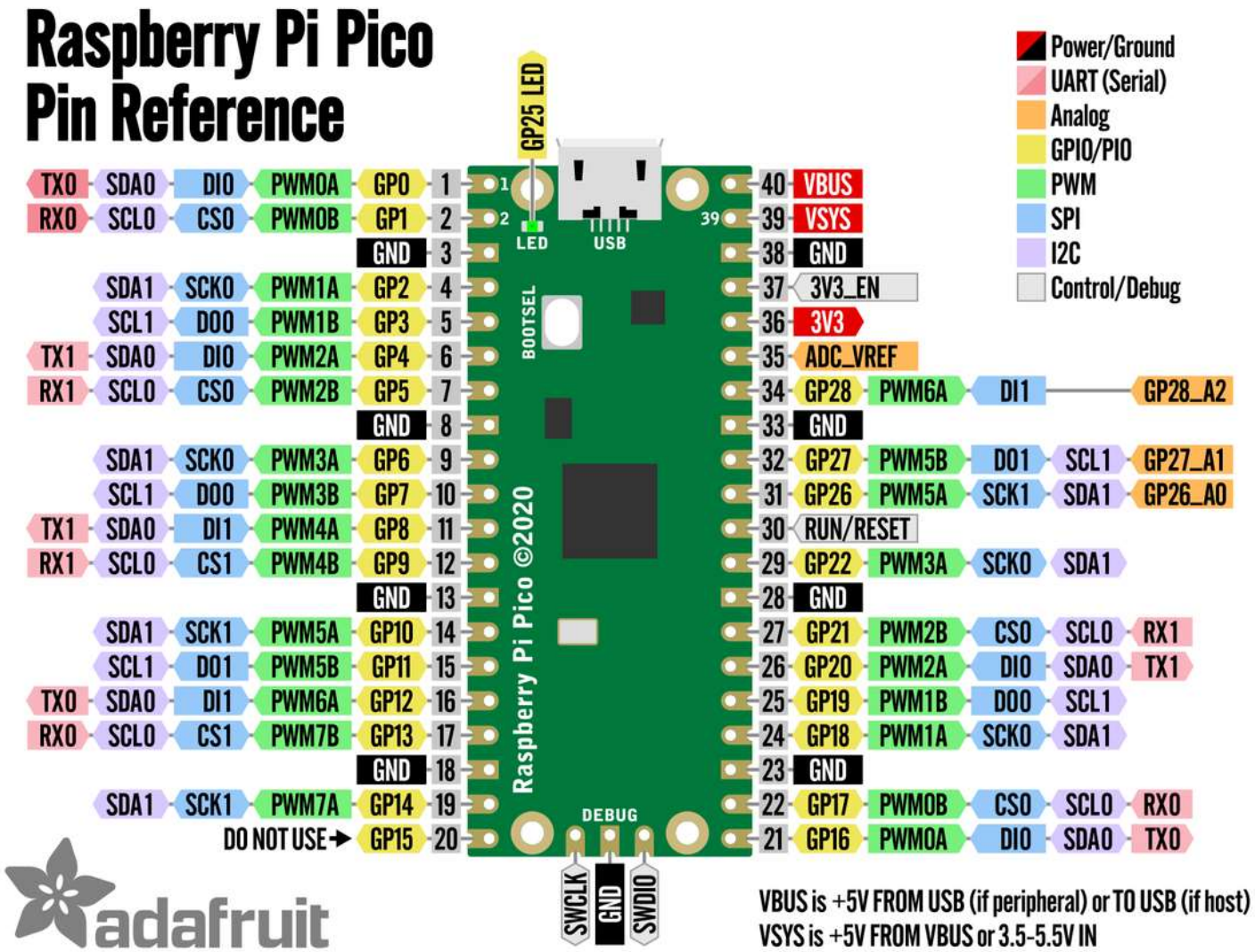
Mouli Sankaran

Lecture 3.4.3 Focus

- DMA in RP2040
- DMA Explained

DMA in RP2040

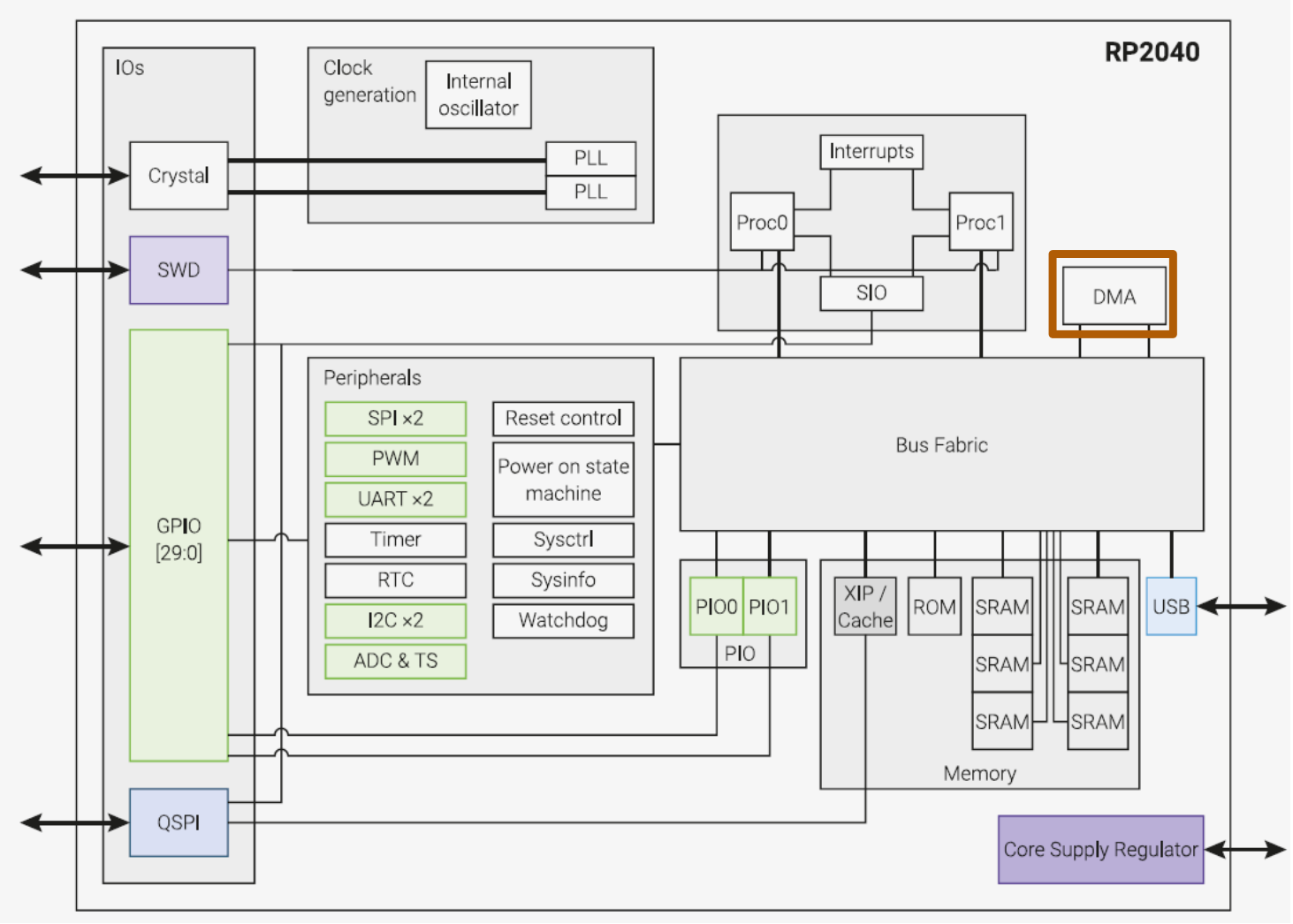
RP2040 Pin Out: Will you find DMA Signals here?



No. In RP2040 DMA is configured to be used within the built-in peripherals and memories.

Reason: RP2040 Meant for low power Applications and bringing DMA signals out would increase the pins and its power budget and size too.

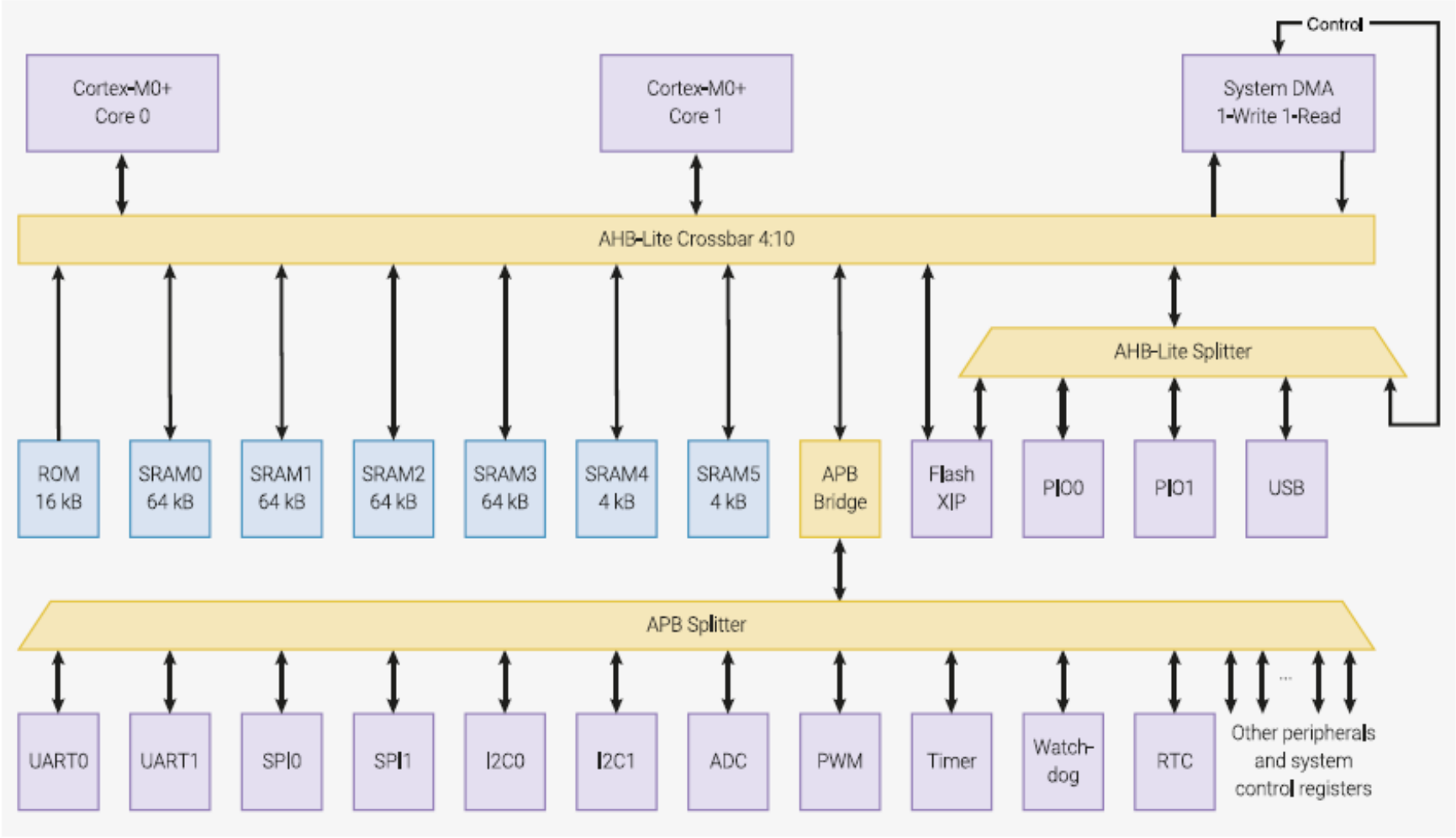
DMA in RP2040 System Block



DMA: Direct Memory Access

Remember, **DMA** has **two master ports** on bus fabric, equivalent to both the cores!!!

DMA Interface with the Bus Fabric



Ref 5: RP2040 Data sheet

Page: 16/647, Figure 4

RP2040: Address Mapping of Peripherals and DMA

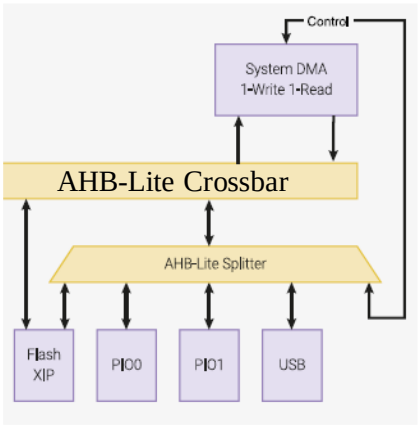
ROM	0x00000000
XIP	0x10000000
SRAM	0x20000000
APB Peripherals	0x40000000
AHB-Lite Peripherals	0x50000000
IOPORT Registers	0xd0000000
Cortex-M0+ internal registers	0xe0000000

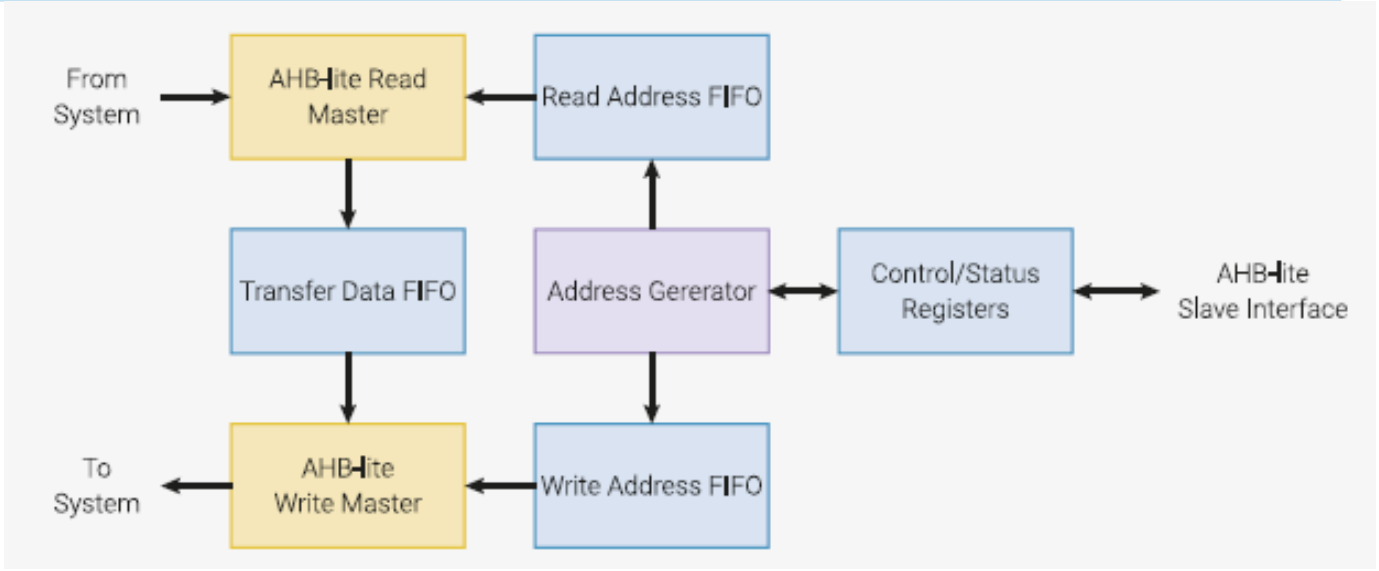
AHB-Lite peripherals:

DMA_BASE	0x50000000

DMA: Block

Ref 5: RP2040 Data sheet
Page: 93/647, Figure 12 DMA





- DMA controller has separate read and write master connections to the bus fabric, and performs bulk data transfers on a processor's behalf.
- This leaves processors free to attend to other tasks, or enter low-power sleep states.
- The data throughput of the DMA is also significantly higher than RP2040 cores.

DMA: Explained

Ref 5: RP2040 Data sheet
Page: 93/647, Figure 12 DMA

- The DMA can perform one read access and one write access, up to 32 bits in size, every clock cycle.
- There are **12 independent channels**, each which supervise a sequence of bus transfers, usually in one of the following scenarios:
  - **Memory-to-peripheral**: a peripheral signals the DMA when it needs more data to transmit. The DMA reads data from an array in RAM or flash, and writes to the peripheral's data FIFO.
  - **Peripheral-to-memory**: a peripheral signals the DMA when it has received data. The DMA reads this data from the peripheral's data FIFO, and writes it to an array in RAM.
  - **Memory-to-memory**: the DMA transfers data between two buffers in RAM, as fast as possible.
- Each channel has its own control and status registers (CSRs), with which software can program and monitor the channel's progress.
- When multiple channels are active at the same time, the DMA shares bandwidth evenly between the channels, with round-robin over all channels which are currently requesting data transfers.

DMA: More Details

Ref 5: RP2040 Data sheet
Page: 93/647, Figure 12 DMA

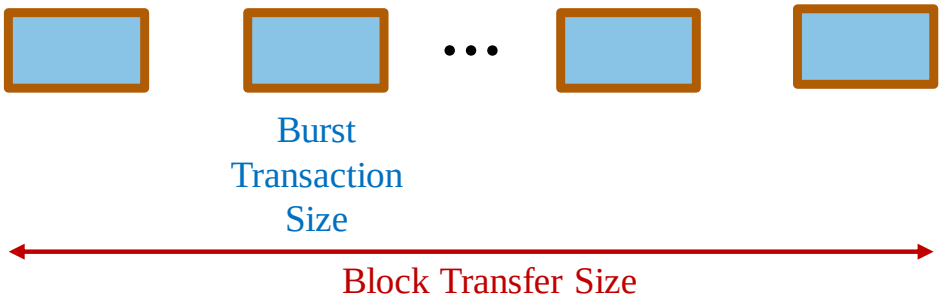
- The transfer size can be either 32, 16, or 8 bits.
- This is configured once per channel: source transfer size and destination transfer size are the same.
- The DMA performs standard byte lane replication on narrow writes, so byte data is available in all 4 bytes of the data bus, and halfword data in both halfwords.
- Channels can be combined in varied ways for more sophisticated behaviour and greater autonomy.
- Making the DMA more autonomous means that much less processor supervision is required: overall this allows the system to do more at once, or to dissipate less power

DMA: Channel Configuration

- Each channel has four control/status registers:
  - **READ_ADDR** is a pointer to the next address to be read from
  - **WRITE_ADDR** is a pointer to the next address to be written to
  - **TRANS_COUNT** shows the number of transfers remaining in the current transfer sequence, and is used to program the number of transfers in the next transfer sequence).
  - **CTRL** is used to configure all other aspects of the channel's behaviour, to enable/disable it, and to check for completion.
- These are **live registers**: they update continuously as the channel progresses
- Read and write address registers update automatically after each read/write access.
- They increment by 1, 2 or 4 bytes at a time, depending on the transfer size configured in CTRL.
- Software should generally program these registers with new start addresses each time a new transfer sequence starts.

Ref 5: RP2040 Data sheet
Page: 93/647, Section 2.5.1

DMA: Block Transfer and Burst Transaction Sizes



Between a **source** and a **destination**

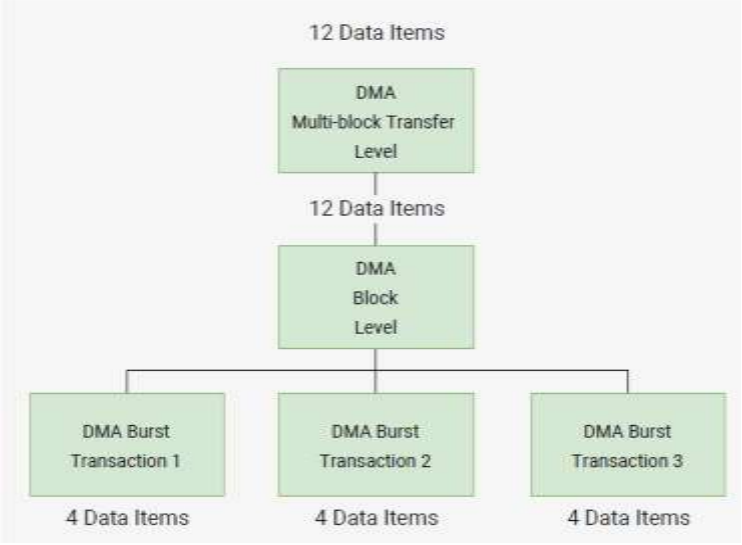
Source and destination can be any part of memory and peripherals, which are provided with DMA transfer capability

Example: UART, USB, etc.

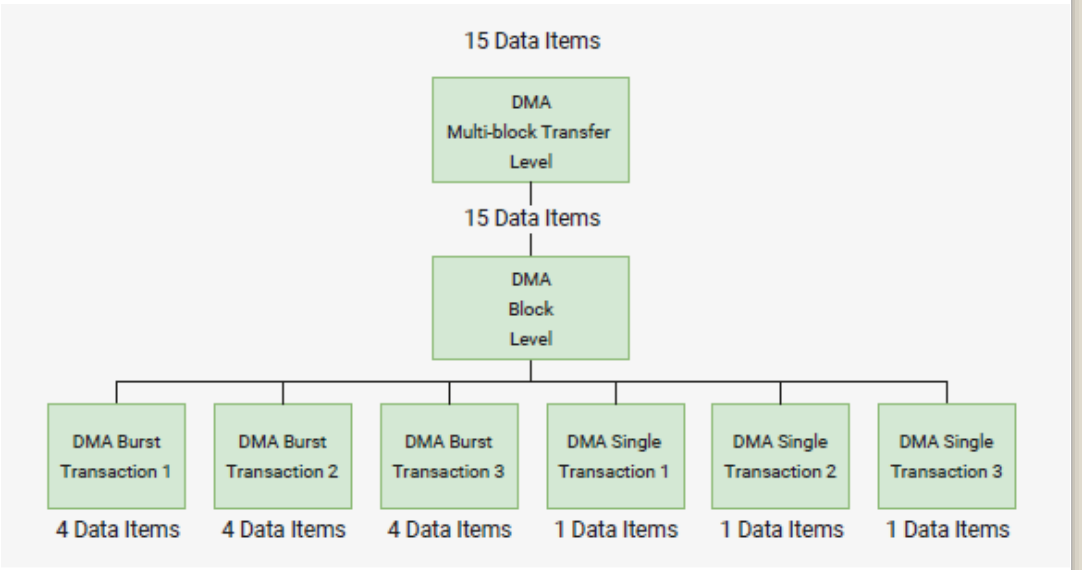
- The DMA Controller is programmed by the processor with the number of data items (block size) that are to be transmitted or received on every **block transfer**.
- The block is broken into a number of transactions.
- The DMA Controller must also be programmed with the number of data items to be transferred for **each DMA request**.
- This is known as the **burst transaction length**.
- Both the sizes are programmed while configuring the DMA for a transfer between chosen pair of devices (memory and/or peripherals) within RP2040.

Examples: DMA Transactions

- Total Block Transfer size : **12 Bytes**
- Burst Transaction size : **4 Bytes**



- Total Block Transfer size : **15 Bytes**
- Burst Transaction size : **4 Bytes**



Note: Burst transaction size is set to 4 bytes because AHB-lite fabric has 32-bits wide interconnections between the masters and slaves.

DMA IRQ

Ref 5: RP2040 Data sheet
Page: 61/647, Table 80 Interrupts

IRQ	Interrupt Source	IRQ	Interrupt Source	IRQ	Interrupt Source	IRQ	Interrupt Source	IRQ	Interrupt Source
0	TIMER_IRQ_0	6	XIP_IRQ	12	DMA_IRQ_1	18	SPI0_IRQ	24	I2C1_IRQ
1	TIMER_IRQ_1	7	PIO0_IRQ_0	13	IO_IRQ_BANK0	19	SPI1_IRQ	25	RTC_IRQ
2	TIMER_IRQ_2	8	PIO0_IRQ_1	14	IO_IRQ_QSPI	20	UART0_IRQ		
3	TIMER_IRQ_3	9	PIO1_IRQ_0	15	SIO_IRQ_PROC0	21	UART1_IRQ		
4	PWM_IRQ_WRAP	10	PIO1_IRQ_1	16	SIO_IRQ_PROC1	22	ADC_IRQ_FIFO		
5	USBCTRL_IRQ	11	DMA_IRQ_0	17	CLOCKS_IRQ	23	I2C0_IRQ		

- Each core is equipped with a standard ARM Nested Vectored Interrupt Controller (NVIC) which has 32 interrupt inputs.
- Each NVIC has the same interrupts routed to it, with the exception of the GPIO interrupts.
- For the DMA, Any combination of channel interrupt requests can be routed to either system **DMA_IRQ 1** or **2**.

Summary

- DMA in RP2040
- DMA Explained