# Device X for Workshop

## Device Description

Device X is a simple, user-friendly device intended for use in workshop settings for educational purposes. The primary function of Device X is to perform an operation between two input values and generate a corresponding result. This result is then stored within the device for later retrieval and analysis.

## Functional Description

### Mailbox

The mailbox feature of the device operates by receiving input values and storing them in a designated data_buffer field in the reg_data_buffer register. Once the value is stored, it can be transferred to a specific instance_X field when the corresponding address is set in the offset field inside the reg_address register. This process involves the copying of data from the data_buffer field to the instance_X field via the reg_buffer register.

### Operation

The operation feature is designed to execute a specific operation (e.g., addition, subtraction, multiplication) between two values stored in the instance_X field of the reg_buffer register. The operation to be performed is dictated by the command given in the command register. Upon completion of the operation, the result is stored in the output field in the reg_output register for subsequent use or analysis.

### Register Definition

#### B1 Registers

Table: DEVICE X REGISTER BLOCK

| Name | Address Offset | Feature |
|---|---|---|
| reg_buffer_size | 0x0 | |
| reg_ *data_buffer* | 0x4 | Mailbox |
| reg_address | 0x8 | Mailbox |
| reg_buffer | 0x20 | Mailbox/Operation |
| reg_command | 0x40 | Operation |
| reg_output | 0x60 | Operation |

reg_buffer_size

reg_buffer_size: 0x0

| Field Name | Field Access | Field Width | Field Offset | Field Default | Field Description |
|---|---|---|---|---|---|
| **size** | RO | 64 | 0 | 0 | |

reg_data_buffer
reg_data_buffer: 0x4

| Field Name | Field Access | Field Width | Field Offset | Field Default | Field Description |
|---|---|---|---|---|---|
| **data_buffer** | RW | 64 | 0 | 0 | Stores the data to be copied into the reg_buffer.instance_x |

reg_address
reg_address: 0x8

| Field Name | Field Access | Field Width | Field Offset | Field Default | Field Description |
|---|---|---|---|---|---|
| **offset** | RW | 64 | 0 | 0 | Stores the reg_buffer.instance_x offset. When this register receives a write it will copy the value to the from the reg_data_buffer.data_buffer to a specific reg_buffer.instance_x, based in the offset value |

reg_buffer
reg_buffer: 0x20

| Field Name | Field Access | Field Width | Field Offset | Field Default | Field Description |
|---|---|---|---|---|---|
| **instance_0** | RO | 8 | 0 | 0 | |
| **instance_1** | RO | 8 | 8 | 0 | |
| **instance_2** | RO | 8 | 16 | 0 | |
| **instance_3** | RO | 8 | 14 | 0 | |

reg_command
reg_command: 0x40

| Field Name | Field Access | Field Width | Field Offset | Field Default | Field Description |
|---|---|---|---|---|---|
| **cmd** | RW | 8 | 0 | 0 | When this field receives a value, it will operate the fields from reg_buffer register according to the encoding and should store the result in the reg_output.output field<br>001: instance_0 +instance_1<br>010: instance_1 - instance_2 |

| | | | | | |
|---|---|---|---|---|---|
| | | | | | 100: instance_2 * instance_3<br>Any other value is Reserved. Illegal programing. |

reg_output

reg_output: 0x60

| Field Name | Field Access | Field Width | Field Offset | Field Default | Field Description |
|---|---|---|---|---|---|
| **output** | RO | 64 | 0 | 0 | This field will allocate the result from the cmd operation |