# rtf6829a

The 6829a allows mapping of addresses for up to 32 tasks per core instance. Addresses are mapped in 2kiB blocks into a 16MB address range. Up to eight mmu cores may be used in a system, allowing up to 256 tasks to be mapped.

# Registers:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Reg | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | Maps |  |
| 00 | WP |  |  | PA23 | PA22 | PA21 | PA20 | PA19 | $000-$7FF |  |
| 01 | PA18 | PA17 | PA16 | PA15 | PA14 | PA13 | PA12 | PA11 |  |
| 02 | WP |  |  | PA23 | PA22 | PA21 | PA20 | PA19 | $800-$FFF |  |
| 03 | PA18 | PA17 | PA16 | PA15 | PA14 | PA13 | PA12 | PA11 |  |
| 04 | WP |  |  | PA23 | PA22 | PA21 | PA20 | PA19 | $1000-$17FF |  |
| 05 | PA18 | PA17 | PA16 | PA15 | PA14 | PA13 | PA12 | PA11 |  |
| 06-3D | … | | | | | | | |  |  |
| 3E | WP |  |  | PA23 | PA22 | PA21 | PA20 | PA19 | $F800-$FFFF |  |
| 3F | PA18 | PA17 | PA16 | PA15 | PA14 | PA13 | PA12 | PA11 |  |
| 40 |  | | | |  | KV MMU0 | | | Only one key value register is present per mmu instance. |  |
| 41 |  | KV MMU1 | | |  |
| 42 |  | KV MMU2 | | |  |
| 43 |  | KV MMU3 | | |  |
| 44 |  | KV MMU4 | | |  |
| 45 |  | KV MMU5 | | |  |
| 46 |  | KV MMU6 | | |  |
| 47 |  | KV MMU7 | | |  |
| 48 |  |  | | S | 1=System |  |
| 49 |  | Fuse | | |  |  |
| 4A | Access Key | | | | | | | | Task accessed |  |
| 4B | Operate Key | | | | | | | | Current task |  |
| 4C-7F | Undefined | | | | | | | |  |  |

The key value register identifies which mmu is being accessed.

The top three bits of the operate key must match the key value register for that task to be actively mapping addresses. IF there is not a match then the mmu will output all zeros for the physical address. This allows multiple mmu’s to be wire or’d.

The top three bits of the access key must match the key value register in order to access the mmu registers for read or write. The low order five bits of the access key select the map for the task. Also the ‘s’ bit must be set in order to be able to update registers.

The ‘S’ bit is set only as a result of an interrupt (hardware or software) and registers can only be updated if the ‘S’ bit is set.

# Ports

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | width | Dir. |  |  |
| Reset\_n | 1 | I | Active low reset |  |
| num | 3 | I | MMU number |  |
| clk | 1 | I | Clock |  |
| dma | 1 | I | Indicates that a DMA cycle is active |  |
| Rw\_n | 1 | I | 1=read, 0=write |  |
| dbi | 8 | I | Data bus input |  |
| dbo | 8 | O | Databus output |  |
| adr | 16 | I | ‘virtual’ Address input |  |
| Padr\_o | 24 | O | Physical address output |  |
| Wp\_o | 1 | O | Write protect signal output |  |
| Inta | 1 | I | Interrupt acknowledge cycle is active |  |
|  |  |  |  |  |

# Parameters

|  |  |  |  |
| --- | --- | --- | --- |
|  | Size | Default |  |
| pIOAddress | 16 | $F800 | Sets the I/O address range that the core responds to. The core requires a block of 128 addresses. The I/O address parameter must be 128 byte aligned. |
| pInterruptWrites | 4 | 7 | Number of consecutive write cycles that identifies an interrupt |
|  |  |  |  |
|  |  |  |  |