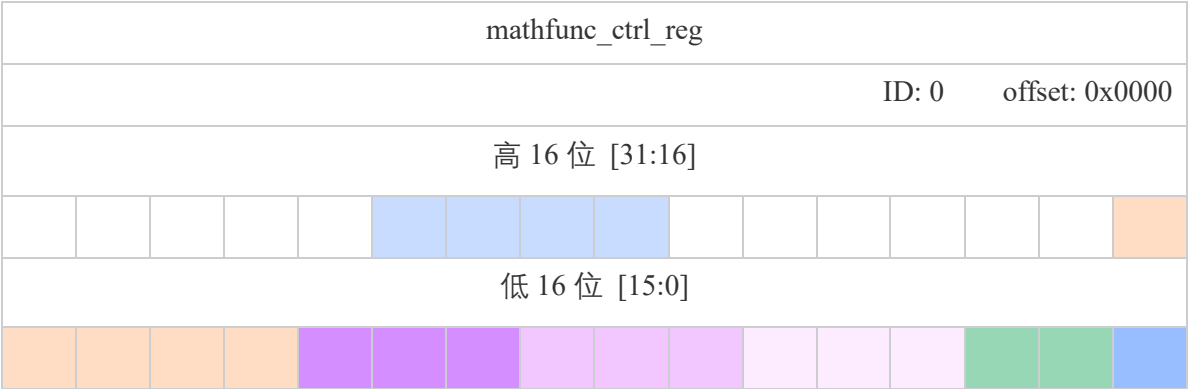


# ncnnAccel RegMap

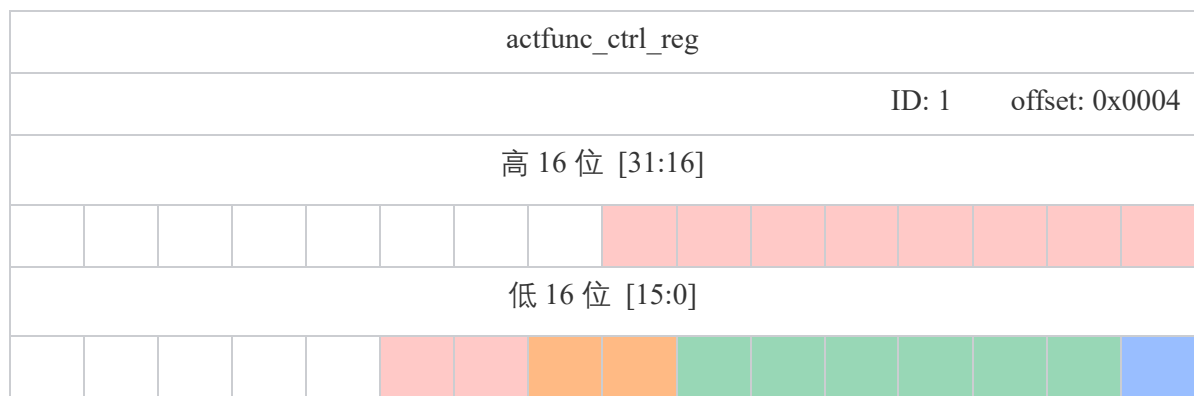
## ALU 寄存器

alu 寄存器组中包含了 mathFunc 和 activationFunc 的配置信号。



- [0:0] ALU mathFunc 运算使能 高电平有效
- [2:1] ALU mathFunc 运算 操作数个数
  - 1: src\_num = 1
  - 2: src\_num = 2
  - 其他: 无效
- [5:3] ALU mul\_src1\_sel
  - 1: dma\_i\_data1
  - 2: math\_alpha
  - 其他: 无效
- [8:6] ALU add\_src0\_sel
  - 1: dma\_i\_data0
  - 2: mul\_o
  - 其他: 无效
- [11:9] ALU add\_src1\_sel
  - 1: dma\_i\_data1
  - 2: beta
  - 其他: 无效
- [12:12] ALU sub\_en
- [13:13] ALU add\_en
- [14:14] ALU mul\_en
- [15:15] ALU max\_en

- [16:16] ALU min\_en
- [23:26] ALU op
  - 1: abs\_en
  - 2: threshold
  - 3: equal\_en
  - 其他: 无效



- [0:0] ALU activationFunc 运算使能 高电平有效
- [2:1] ALU activationFunc 激活函数性质
  - 00:monotonic function
  - 01:odd function
  - 10:even function
- [4:3] ALU activationFunc 运算 输入数据来源
  - 1: from dma controller
  - 2: from oscaleAndBias Module
  - 其他: 无效
- [6:5] ALU activationFunc 运算 输出数据去向
  - 1: to alu dma controller
  - 2: to opfusion Module
  - 其他: 无效
- [23:16] act\_op

以下寄存器均为 32 位宽 (input)

| name               | ID | offset | description   |
|--------------------|----|--------|---------------|
| alu_veclen_ch0_reg | 2  | 0x0008 | CH0 向量长度      |
| src0_addr_ch0_reg  | 3  | 0x000C | CH0 源操作数 0 地址 |

|                      |    |        |                             |
|----------------------|----|--------|-----------------------------|
| src1_addr_ch0_reg    | 4  | 0x0010 | CH0 源操作数 1 地址               |
| dst_addr_ch0_reg     | 5  | 0x0014 | CH0 目的地址                    |
| alu_veclen_ch1_reg   | 6  | 0x0018 | CH1 向量长度                    |
| src0_addr_ch1_reg    | 7  | 0x001C | CH1 源操作数 0 地址               |
| src1_addr_ch1_reg    | 8  | 0x0020 | CH1 源操作数 1 地址               |
| dst_addr_ch1_reg     | 9  | 0x0024 | CH1 目的地址                    |
| math_alpha_reg       | 10 | 0x0028 | 用于计算 $Y=\alpha * X + \beta$ |
| math_beta_reg        | 11 | 0x002C | 用于计算 $Y=\alpha * X + \beta$ |
| act_range_reg[0]     | 12 | 0x0030 | range x[0]                  |
| act_range_reg[1]     | 13 | 0x0034 | range x[1]                  |
| act_range_reg[2]     | 14 | 0x0038 | range x[2]                  |
| act_range_reg[3]     | 15 | 0x003C | range x[3]                  |
| act_coefficient_a[0] | 16 | 0x0040 | coefficient : a             |
| act_coefficient_a[1] | 17 | 0x0044 | coefficient : a             |
| act_coefficient_a[2] | 18 | 0x0048 | coefficient : a             |
| act_coefficient_a[3] | 19 | 0x004C | coefficient : a             |
| act_coefficient_a[4] | 20 | 0x0050 | coefficient : a             |
| act_coefficient_b[0] | 21 | 0x0054 | coefficient : b             |
| act_coefficient_b[1] | 22 | 0x0058 | coefficient : b             |
| act_coefficient_b[2] | 23 | 0x005C | coefficient : b             |
| act_coefficient_b[3] | 24 | 0x0060 | coefficient : b             |
| act_coefficient_b[4] | 25 | 0x0064 | coefficient : b             |
| act_coefficient_c[0] | 26 | 0x0068 | coefficient : c             |
| act_coefficient_c[1] | 27 | 0x006C | coefficient : c             |
| act_coefficient_c[2] | 28 | 0x0070 | coefficient : c             |
| act_coefficient_c[3] | 29 | 0x0074 | coefficient : c             |
| act_coefficient_c[4] | 30 | 0x0078 | coefficient : c             |

---

|                           |
|---------------------------|
| innerproductFunc_ctrl_reg |
|---------------------------|

|                            |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|----------------------------|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
| ID: 31      offset: 0x007C |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 高 16 位 [31:16]             |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|                            |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 低 16 位 [15:0]              |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|                            |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

- [0:0] ALU innerproductFunc 运算使能 高电平有效

以下寄存器均为 32 位宽 (output)

| name          | ID | offset | description       |
|---------------|----|--------|-------------------|
| alu_odata_reg | 34 | 0x0088 | 用于 sum 等运算的单个数据返回 |

## 池化寄存器

pool 寄存器用于描述池化单元的相关控制信号。

|                            |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|----------------------------|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
| pool_ctrl_reg              |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| ID: 40      offset: 0x00A0 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 高 16 位 [31:16]             |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|                            |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 低 16 位 [15:0]              |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|                            |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

- [0:0] 池化运算使能 高电平有效
- [2:1] 池化类型
  - 0: 均值池化
  - 1:最大值池化
  - 其他:保留
- [4:3] kernel\_w
- [6:5] kernel\_h
- [8:7] stride\_w
- [10:9] stride\_h

- [12:11] pool 单元数据输入来源
  - 1: from dma
  - 2: from activation Module
  - 其他: 保留
- [17:16] pad\_bottom
- [19:18] pad\_top
- [21:20] pad\_right
- [23:22] pad\_left
- [24:24] pad\_mode
  - 0: padding const // 补 pad\_value\_reg 的值
  - 1: padding edge

以下寄存器均为 32 位宽 (input)

| name                  | ID | offset | description                          |
|-----------------------|----|--------|--------------------------------------|
| pool_shape_ic_reg     | 41 | 0x00A4 | 输入特征图的 ic, 低 16 位有效                  |
| pool_shape_iwh_reg    | 42 | 0x00A8 | 输入特征图的 iwh<br>高 16 位为 iw, 低 16 位为 ih |
| pool_shape_icstep_reg | 43 | 0x00AC | 输入特征图的 cstep, 用于地址对齐                 |
| pool_shape_oc_reg     | 44 | 0x00B0 | 输出特征图的 oc, 低 16 位有效                  |
| pool_shape_owh_reg    | 45 | 0x00B4 | 输出特征图的 owh<br>高 16 位为 ow, 低 16 位为 oh |
| pool_shape_ocstep_reg | 46 | 0x00B8 | 输出特征图的 cstep, 用于地址对齐                 |
| pool_ifm_addr_reg     | 47 | 0x00BC | 输入特征图地址                              |
| pool_ofm_addr_reg     | 48 | 0x00C0 | 输出特征图地址                              |
| pool_pad_value_reg    | 49 | 0x00C4 | pad_mode = 0 时 pad_value 的值          |

## GEMM 寄存器

|                            |
|----------------------------|
| gemm_ctrl_reg              |
| ID: 60      offset: 0x00F0 |



- [0:0] gemm 单元使能 高电平有效
- [2:1] gemm 加速类型
  - 0: 卷积
  - 1: 深度卷积
  - 2: 矩阵乘
  - 3: 保留
- [5:3] 卷积核长宽尺寸,  $\text{kernel\_w} = \text{kernel\_h}$
- [8:6] 卷积核步进,  $\text{stride\_w} = \text{stride\_h}$
- [10:9] padding 模式
  - 0: 填充 0
  - 1: 填充边缘数据 (尚未支持)
- [12:11] left 侧 padding 长度
- [14:13] right 侧 padding 长度
- [16:15] top 侧 padding 长度
- [18:17] bottom 侧 padding 长度
- [19:19] bias 使能, 高电平有效
- [20:20] 重量化使能, 高电平有效
- [21:21] layout 使能, 高电平有效
- [22:22] oscale 使能, 高电平有效
- [23:23] div\_ifm\_c\_en 权重缓冲区或 oscaleAndBias 缓冲区不足, 引起的 ic 分块

| name             | ID | offset | description  |
|------------------|----|--------|--------------|
| quant_data_reg   | 61 | 0x00F4 | 量化因子, 单精度浮点  |
| requant_data_reg | 62 | 0x00F8 | 重量化因子, 单精度浮点 |
| dequant_addr_reg | 63 | 0x00FC | 反量化因子地址      |
| bias_addr_reg    | 64 | 0x0100 | 偏置地址         |

|                     |    |        |                                     |
|---------------------|----|--------|-------------------------------------|
| ifm_shape_c_reg     | 65 | 0x0104 | 输入特征图的 ic                           |
| ifm_shape_wh_reg    | 66 | 0x0108 | 输入特征图的 iwh<br>高 16 位为 iw，低 16 位为 ih |
| ifm_shape_cstep_reg | 67 | 0x010C | 输入特征图的 cstep，用于地址对齐                 |
| ofm_shape_c_reg     | 68 | 0x0110 | 输入特征图的 oc                           |
| ofm_shape_wh_reg    | 69 | 0x0114 | 输入特征图的 owh<br>高 16 位为 ow，低 16 位为 oh |
| ofm_shape_cstep_reg | 70 | 0x0118 | 输入特征图的 cstep，用于地址对齐                 |
| wgt_len_reg         | 71 | 0x011C | 权重长度                                |
| ifm_baseaddr_reg    | 72 | 0x0120 | 输入特征图基地址                            |
| wgt_baseaddr_reg    | 73 | 0x0124 | 权重基地址                               |
| ofm_baseaddr_reg    | 74 | 0x0128 | 输出特征图基地址                            |
| div_ifm_c_reg       | 75 | 0x012C | oc 分块时，ic 对齐到 32 的值                 |