LeNet-5 ISA Gen Example

System – Configuration/Definition

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Configuration | | Definition | | Description |
| DRAM\_BASE\_ADDR | 0x50000000 | LAYER\_CONFIG | 4’d0 | ISA Type |
| IFM\_INIT\_ADDR | 0x5e000000 | TRANS\_CONFIG | 4’d1 |
| OFM\_INIT\_ADDR | 0x5e000000 | LOAD | 4’d2 |
| WT\_INIT\_ADDR | 0x5f000000 | STORE | 4’d3 |
|  |  | CONV | 4’d4 |
|  |  | POOLING | 4’d5 |
|  |  | SRAM\_ID\_IFM | 2’d0 | Local SRAM ID |
|  |  | SRAM\_ID\_WT | 2’d1 |
|  |  | SRAM\_ID\_OFM | 2’d2 |
|  |  | ACT\_RELU | 0x0 | Activation Type |
|  |  | ACT\_PRELU | 0x1 |
|  |  | ACT\_NO | 0x2 |

Layer0 – Configuration

|  |  |  |  |
| --- | --- | --- | --- |
| channel | 1 | left\_pad | 0 |
| size | 2 | right\_pad | 0 |
| filter | 6 | last | 1 |
| kernel | 3 | act\_type | ACT\_RELU |
| stride | 1 | pool\_size | 2 |
| shift\_bits | 9 | DRAM\_WT\_ADDR | WT\_INIT\_ADDR |
| top\_pad | 0 | DRAM\_OFM\_ADDR | OFM\_INIT\_ADDR |
| bot\_pad | 0 |  |  |

![一張含有 桌 的圖片

自動產生的描述]()

Layer0 – ISA Flow

|  |  |  |  |
| --- | --- | --- | --- |
| Type | Value | Description | Count |
| Layer | {7’b0,1’b0, stride, filter, kernel, channel, size, size, LAYER\_CONFIG} | Set layer configuration | 1 |
| Transfer | {22’b0, right\_pad, left\_pad, bot\_pad, top\_pad, channel, size, size, TRANS\_CONFIG} | Load IFM data | 1 |
| Load | {5’b0, 17’b0, SRAM\_ID\_IFM, IFM\_INIT\_ADDR, 4’b0, LOAD} | 1 |
| Transfer | {22’b0, right\_pad, left\_pad, bot\_pad, top\_pad, channel, kernel, kernel, TRANS\_CONFIG} | Load Weight data  Repeat local\_filter times | 6 |
| Load | {5’b0, SRAM\_WT\_ADDR, SRAM\_ID\_WT, DRAM\_WT\_ADDR, 4’b0, LOAD} | 6 |
| Conv | {8’b0, shift\_bits, last, act\_type, 7’b0, 4’b0, 6’b0, size, size, 12’b0, 4’b0, CONV} | Convolution | 1 |
| Transfer | {22’b0, right\_pad, left\_pad, bot\_pad, top\_pad, local\_filter, (size-2), (size-2), TRANS\_CONFIG} | Store result | 1 |
| Store | {5’b0, 17’b0, SRAM\_ID\_OFM, DRAM\_OFM\_ADDR, 4’b0, STORE} | 1 |

* local\_filter代表此次計算會處理多少個filter，因為CIM一次最多8個filter。在Layer0 local filter=6，一個iteration就結束，如上表所示。
* Load Weight data時，SRAM\_WT\_ADDR跟DRAM\_WT\_ADDR要做逐次累加，公式如下:

incr\_addr = ((kernel\*kernel\*channel)%4 == 0) ? kernel\*kernel\*channel

: (kernel\*kernel\*channel/4+0x1)\*4

DRAM\_WT\_ADDR = DRAM\_WT\_ADDR + incr\_addr

SRAM\_WT\_ADDR = SRAM\_WT\_ADDR + kernel\*kernel\*channel