NPU 指令集

下面简单介绍一下 NPU 支持的指令。其中，$1 表示输入参数 1；$2 表示输入参数矩阵 2；$3 表示输出矩阵 3；IMM 表示立即数。x 表示矩阵乘法，.x 表示矩阵点乘，*表示卷积。

助记符	功能	127:124	123:92	91:60	59:28	27:0
ADD	$3=$1+$2	4'B0000	$1 地址	$2 地址	$3 地址	[27:19]为矩阵行数； [18:10]为矩阵列数； [9:0]保留无用
ADDi	$3=$1+IMM	4'B0001	$1 地址	IMM	$3 地址	[27:19]为矩阵行数； [18:10]为矩阵列数； [9:0]保留无用
SUB	$3=$1-$2	4'B0010	$1 地址	$2 地址	$3 地址	[27:19]为矩阵行数； [18:10]为矩阵列数； [9:0]保留无用
SUBi	$3=$1-IMM	4'B0011	$1 地址	IMM	$3 地址	[27:19]为矩阵行数； [18:10]为矩阵列数； [9:0]保留无用
MULT	$3=$1x$2	4'B0100	$1 地址	$2 地址	$3 地址	[27:19]为矩阵 1 行数； [18:10]为矩阵 1 列数； [9:1]为矩阵 2 列数； [0]保留无用
MULTi	$3=$1xIMM	4'B0101	$1 地址	IMM	$3 地址	[27:19]为矩阵行数； [18:10]为矩阵列数； [9:0]保留无用
DOT	$3=$1.x$2	4'B0110	$1 地址	$2 地址	$3 地址	[27:19]为矩阵行数； [18:10]为矩阵列数； [9:0]保留无用
TRAN	$3=($1)' 矩阵转置	4'B1101	$1 地址	保留	$3 地址	[27:19]为矩阵 1 行数； [18:10]为矩阵 1 列数
CONV	$3=$1*$2 $1 为图像 $2 为卷积核	4'B0111	$1 地址	$2 地址	$3 地址	[27:19]为矩阵 1 行数； [18:10]为矩阵 1 列数； [9:5]为矩阵 2 行数； [4:0]为矩阵 2 列数

助记符	功能	127:124	123:92	91:60	59:28	27:0
POOL	$3=down($1) 下采样操作	4'B1000	$1 地址	MODE 0: max 1: mean	$3 地址	[27:19]为矩阵 1 行数； [18:10]为矩阵 1 列数； [9:5]为矩阵 2 行数； [4:0]为矩阵 2 列数
SIGM	$3=sigm($1)	4'B1001	$1 地址	保留	$3 地址	[27:19]为矩阵 1 行数； [18:10]为矩阵 1 列数
RELU	$3=relu($1)	4'B1010	$1 地址	保留	$3 地址	[27:19]为矩阵 1 行数； [18:10]为矩阵 1 列数
TANH	$3=tanh($1)	4'B1011	$1 地址	保留	$3 地址	[27:19]为矩阵 1 行数； [18:10]为矩阵 1 列数
GRAY	$3=gray($1)	4'B1100	$1 地址 RGB565	保留	$3 地址 YCbCr	[27:19]为矩阵 1 行数； [18:10]为矩阵 1 列数

为了实现对 NPU 单元运算启动和运算结束的控制，扩展支持了如下控制指令：

助记符	指令值	指令说明
NOP	128'D0 空操作	一旦 NPU 在指令 RAM 中寻址到了 NOP 操作，说明 RAM 里面该执行的指令都完成了
RESET	128'D1 复位 RAM	表示要重新定制指令 RAM 里面的内容
START	128'D2 启动运算	表示 NPU 要启动 RAM 里指令的运行

PS: 为了 NPU 实现的方便，这里不限定 CONV 卷积运算中卷积核尺寸；POOL 池化运算中，池化核尺寸也没有限定。但是所有矩阵操作都必须满足矩阵大小在 511 x 511 以内。尽管损失了一些灵活性，但是系统结构变得十分的简单。

另外，卷积、池化的实现，依赖于 npu_conv_rtl 模块，这里有参数 Km/Kn，CNN 设计中的卷积/池化尺寸必须在(Km, Kn)内。只要(1,1) ~ (Km, Kn)内任意尺寸都可以