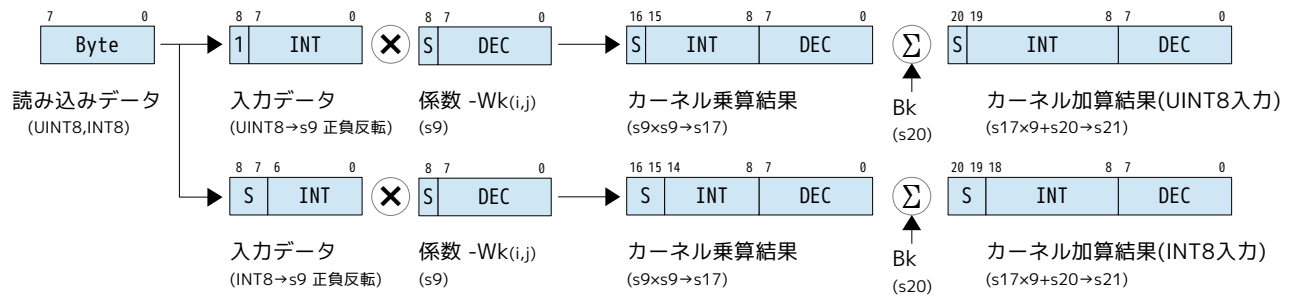
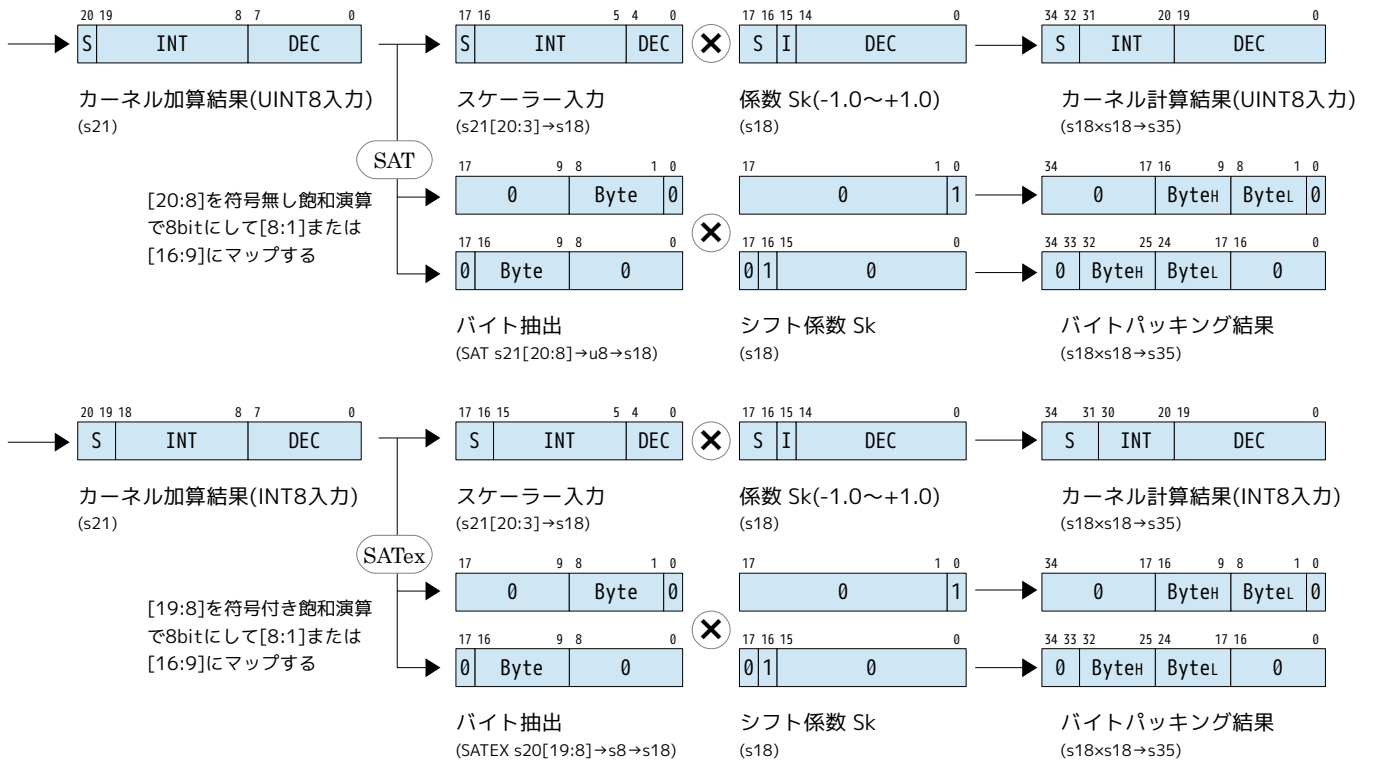


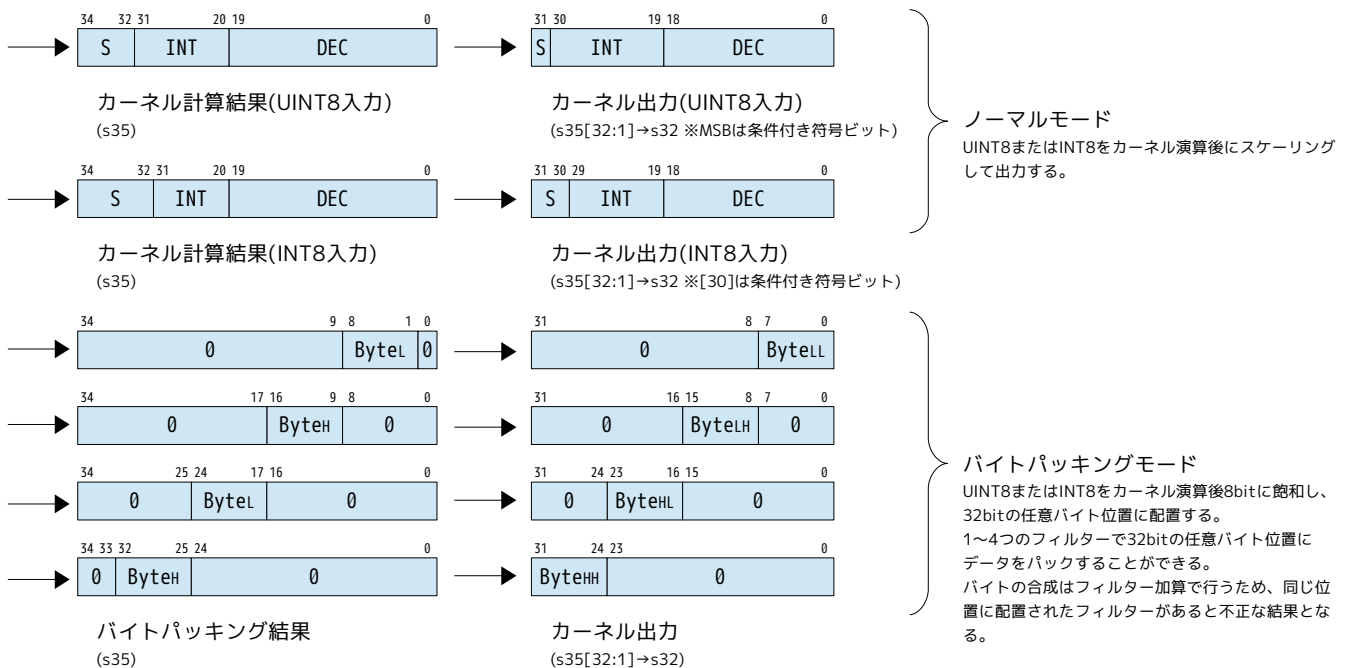
# カーネル演算精度の検討



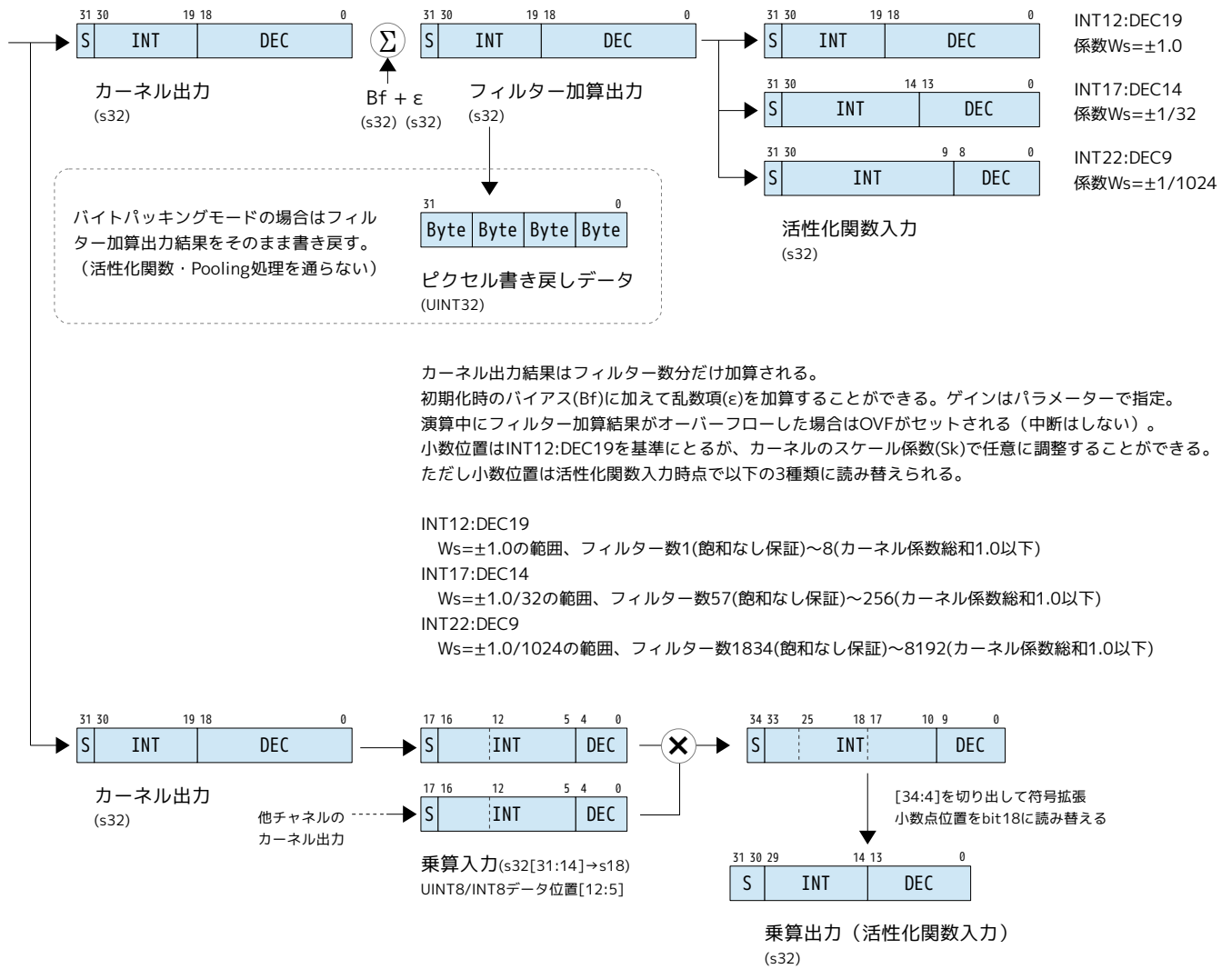
(続き)



(続き)



## フィルター加算および乗算精度の検討



カーネル出力結果はフィルター数分だけ加算される。

初期化時のバイアス(Bf)に加えて乱数項( $\epsilon$ )を加算することができる。ゲインはパラメーターで指定。

演算中にフィルター加算結果がオーバーフローした場合はOVFがセットされる(中断はしない)。

小数位置はINT12:DEC19を基準にとるが、カーネルのスケール係数( $S_k$ )で任意に調整することができる。

ただし小数位置は活性化関数入力時点で以下の3種類に読み替えられる。

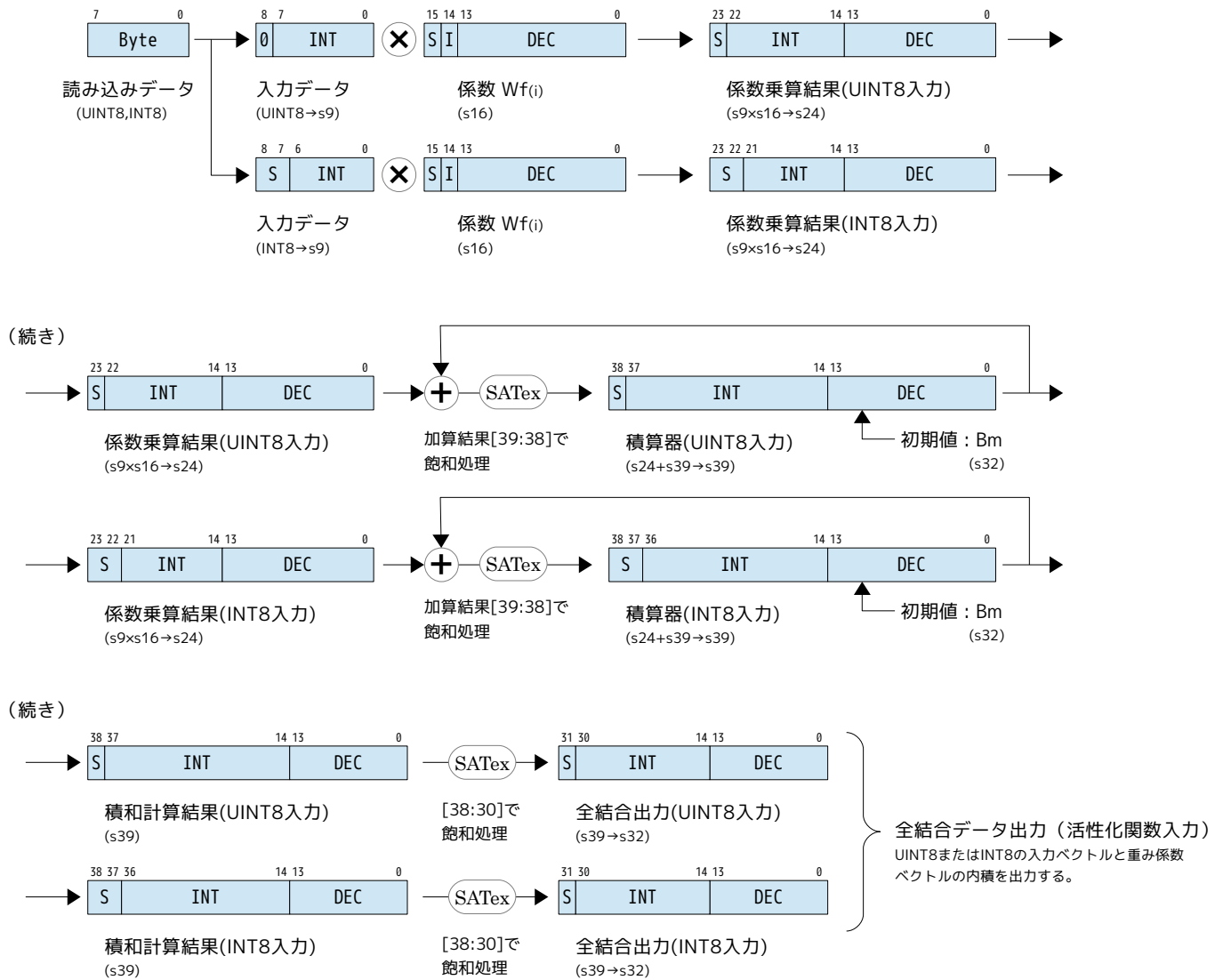
乗算モードの場合、指定された2チャンネルのカーネル出力結果同士を乗算する。

フィルターパラメーター数が2以外の場合の計算結果は不定。

2つの入力のアダマール積を計算するために使用するため、基本的にはUINT8/INT8データのパススルー(1x1カーネル, 1.0倍)を8bit固定小数とみなして小数位置の読み替えを行う。

小数位置はINT17:DEC14を基準にとるが、カーネルのスケール係数( $S_k$ )で任意に調整することができる。

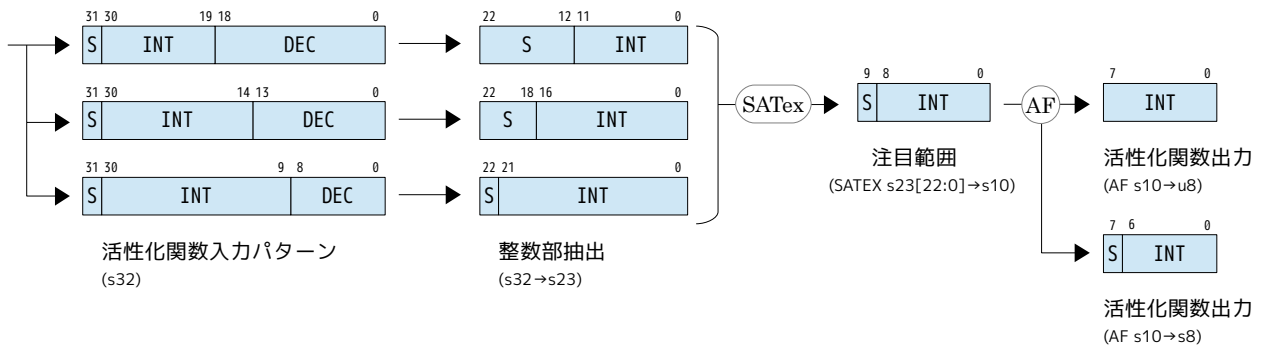
## 全結合演算精度の検討



UINT8/INT8の入力ベクトルとINT16の重みベクトルの内積を計算する。  
 カーネル計算・フィルター加算とは独立してインスタンスされ、出力は活性化関数へ渡される。

積算器の初期値に $Bm$ をとる。 $Bm$ は重みベクトル全体で共通の固定値で、入力ベクトルのゼロ位置オフセットに利用することができる。  
 演算中にフィルター加算結果がオーバーフローした場合はOVFがセットされる（中断はしない）。  
 小数位置はINT17:DEC14を基準にとるが、重み係数をスケーリングすることで任意に調整することができる。

## 活性化関数精度とプーリング処理の検討



活性化関数(AF)

入力された32bit符号付き整数に対して以下の3種類の小数位置から整数部を抽出し、飽和演算で符号付き10bitの注目範囲に丸める。

INT12:DEC19

$Ws = \pm 1.0$ の範囲、フィルター数1(飽和なし保証)～8(カーネル係数総和1.0以下)

INT17:DEC14

$Ws = \pm 1.0/32$ の範囲、フィルター数57(飽和なし保証)～256(カーネル係数総和1.0以下)

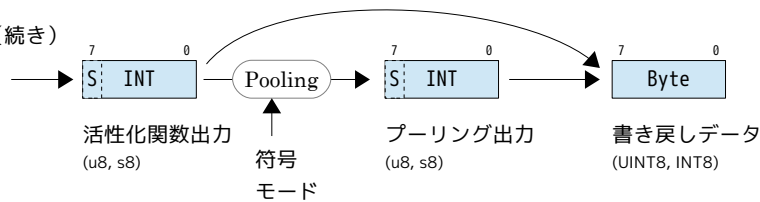
INT22:DEC9

$Ws = \pm 1.0/1024$ の範囲、フィルター数1834(飽和なし保証)～8192(カーネル係数総和1.0以下)

符号付き10bitの値を以下の8種類（後半4種のインスタンスはオプション）の活性化関数で8bitに丸める。

- ・ **ReLU** (UINT8 saturation)  
符号無し飽和演算で0～255のUINT8を出力
- ・ **Hard-tanh** (INT8 saturation)  
符号付き飽和演算で-128～+127のINT8を出力
- ・ **Step**  
符号で0(<0)または255(≥0)のUINT8を出力
- ・ **Leaky-ReLU**  
符号付き飽和演算で-1～+127のINT8を出力  
(負数を-1に丸める)
- ・ **sigmoid/LUT0**  
1024のテーブル参照で0～255のUINT8を出力  
\*Sigmoid関数の飽和範囲は±5.0以上なので、スケーラーで4/5倍にしておく
- ・ **tanh**  
1024のテーブル参照で-128～+127のINT8を出力
- ・ **LUT1**  
1024のテーブル参照で0～255のUINT8を出力
- ・ **LUT2**  
1024のテーブル参照で-128～+127のINT8を出力

(続き)

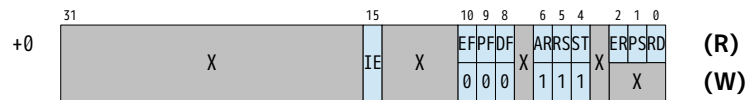


プーリング処理(Pooling)

活性化関数出力を以下の手法で1/4に圧縮する。Pooling処理を行わない場合は活性化関数出力がそのまま書き戻しデータに出力される

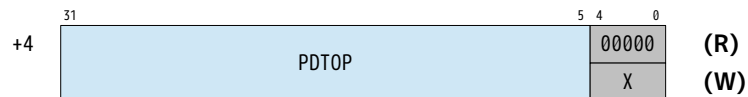
- ・ **MaxPooling**  
2x2のなかで最も大きな値を返す
- ・ **AvgPooling**  
2x2の平均値を返す
- ・ **SimplePooling**  
2x2の左上(0,0)を返す (Conv2DのStride=2と等価)

## PERIDOT\_CNNのレジスタ (1/3)



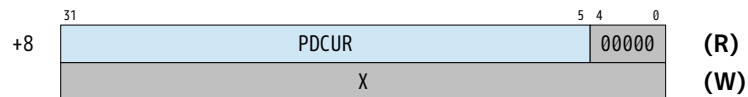
(R)  
(W)

- [0] STAT\_READY : モジュールレディ(RO)  
0 : Busy  
1 : Ready
- [1] STAT\_PAUSE : 一時停止中(RO)  
0 : Normal  
1 : Pause halt (RESTART入力で再開)
- [2] STAT\_ERROR : エラー発生(RO)  
0 : Normal  
1 : Error halt (続行不能)
- [4] START : 処理開始(R/W1) - STAT\_READYアサート時のみ1書き込み可能
- [5] RESTART : 処理再開(R/W1) - STAT\_PAUSEアサート時のみ1書き込み可能
- [6] ABORT : 処理中止(R/W1) - STAT\_READYネゲート時のみ1書き込み可能
- [8] DONE\_IRQ : 処理完了割り込み要因(R/W0) - 0書き込みでクリア
- [9] PAUSE\_IRQ : 一時停止割り込み要因(R/W0) - 0書き込みでクリア
- [10] ERROR\_IRQ : エラー割り込み要因(R/W0) - 0書き込みでクリア
- [15] IRQ\_ENABLE : 割り込みイネーブル  
0 : 割り込みマスク  
1 : DONE\_IRQ,PAUSE\_IRQ,ERROR\_IRQのいずれかがアサートされていたらIRQを発行



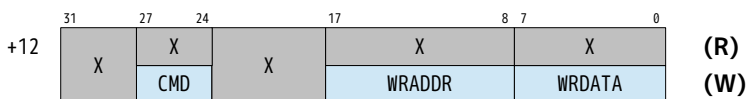
(R)  
(W)

- [31:5] PDTOP : デスクリプタの開始アドレス(32バイト境界に配置)  
STAT\_READYアサート時のみ書き込み可能



(R)  
(W)

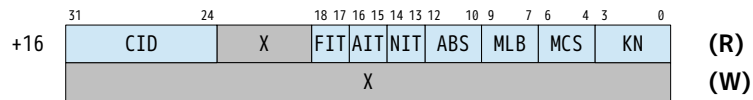
- [31..5] PDCUR : 割り込みを発行したデスクリプタ先頭アドレス(32バイト境界をポイント)  
STAT\_PAUSE,STAT\_ERRORアサート時に有効な値を指示



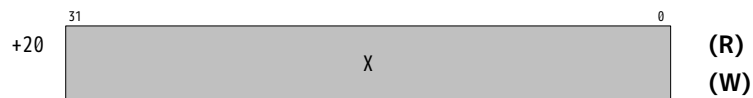
(R)  
(W)

- [7:0] WRDATA : 活性化関数LUTに書き込むデータ(WO)
- [17:8] WRADDR : LUTの書き込み先アドレス(WO)  
0~1023
- [27:24] WRCMD : LUT書き込みコマンド(WO)  
1100 : LUT0に書き込みを実行 (AF\_INST\_TYPE=1,2のとき)  
1010 : LUT1に書き込みを実行 (AF\_INST\_TYPE=3のとき)  
1011 : LUT2に書き込みを実行 (AF\_INST\_TYPE=3のとき)

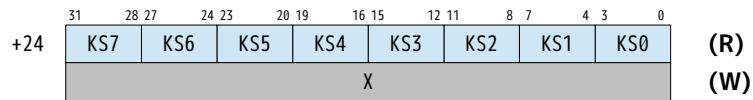
## PERIDOT\_CNNのレジスタ (2/3)



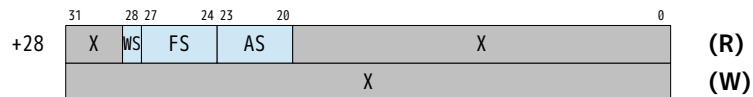
- [3:0] KERNEL\_NUM : インスタンスされているカーネル数(RO)  
1~8
- [6:4] MAXCONV\_SIZE : 畳み込み画像の最大サイズ(RO)  
000 : 256x256 pix  
001 : 512x512 pix  
010 : 1024x1024 pix  
011 : 2048x2048 pix  
100 : 4096x4096 pix  
101~111 : reserved
- [9:7] MAXLINE\_BYTES : 画像1ラインの最大バイト数(RO)  
000 : 1023Bytes  
001 : 2047Bytes  
010 : 4095Bytes  
011 : 8191Bytes  
100 : 16383Bytes  
101 : 32767Bytes  
110~111 : reserved
- [12:10] ACMBUFF\_SIZE : アキュムレーター内部バッファサイズ(RO)  
000 : なし  
001 : 4kBytes (1k words)  
010 : 16kBytes (4k words)  
011 : 64kBytes (16k words)  
100~111 : reserved
- [14:13] NF\_INST\_TYPE : 乱数バイアス関数インスタンス(RO)  
00 : 乱数なし  
01 : 一様乱数, 近似 $\cos^{\wedge}19$   
10~11 : reserved
- [16:15] AF\_INST\_TYPE : 活性化関数インスタンス(RO)  
00 : ReLU, Hard-tanh, Step, Leaky-ReLU  
01 : 00に加えてsigmoid/LUT0  
10 : 01に加えてtanh  
11 : 10に加えてLUT1, LUT2 (※LUT0は無くなる)
- [18:17] FC\_INST\_TYPE : 全結合インスタンス(RO)  
00 : 全結合機能なし  
01 : 整数行列積 (UINT8/INT8データ x INT16重み)  
10~11 : reserved
- [31:24] COMPONENT\_ID : コンポーネントID(RO)  
7Ah : PERIDOT\_CNNコンポーネント Rev.1



## PERIDOT\_CNNのレジスタ (3/3)



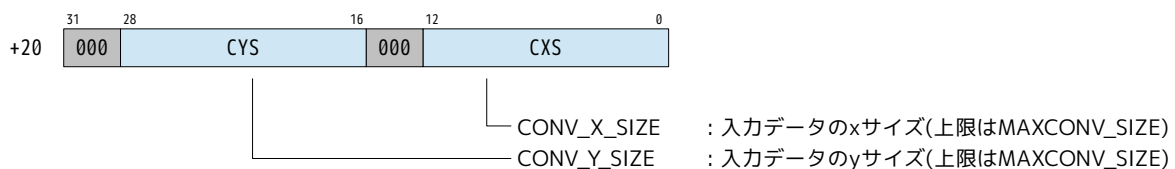
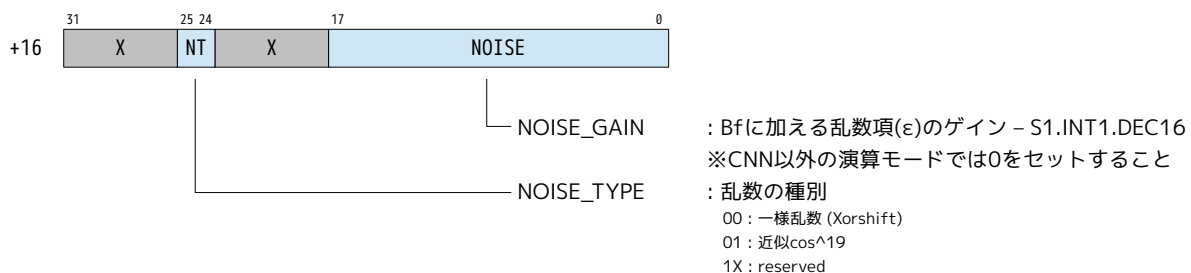
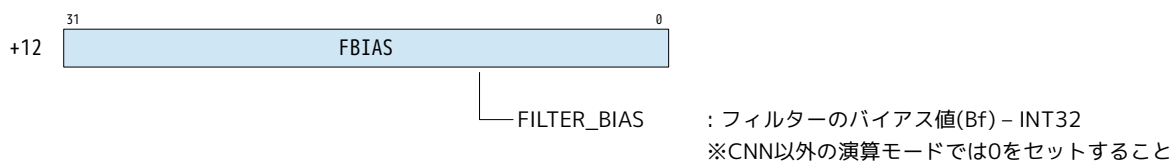
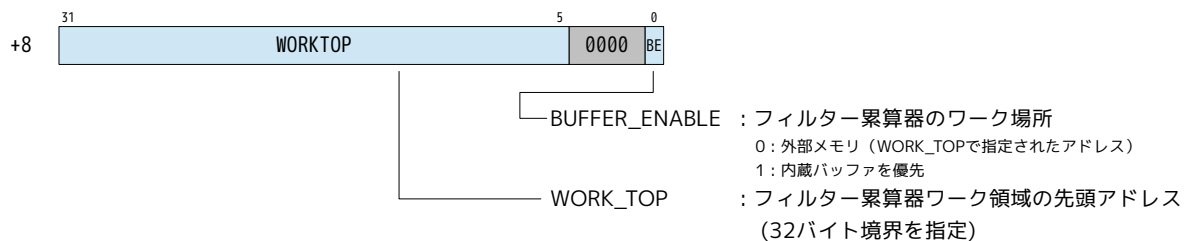
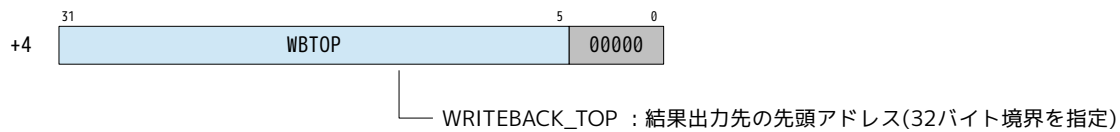
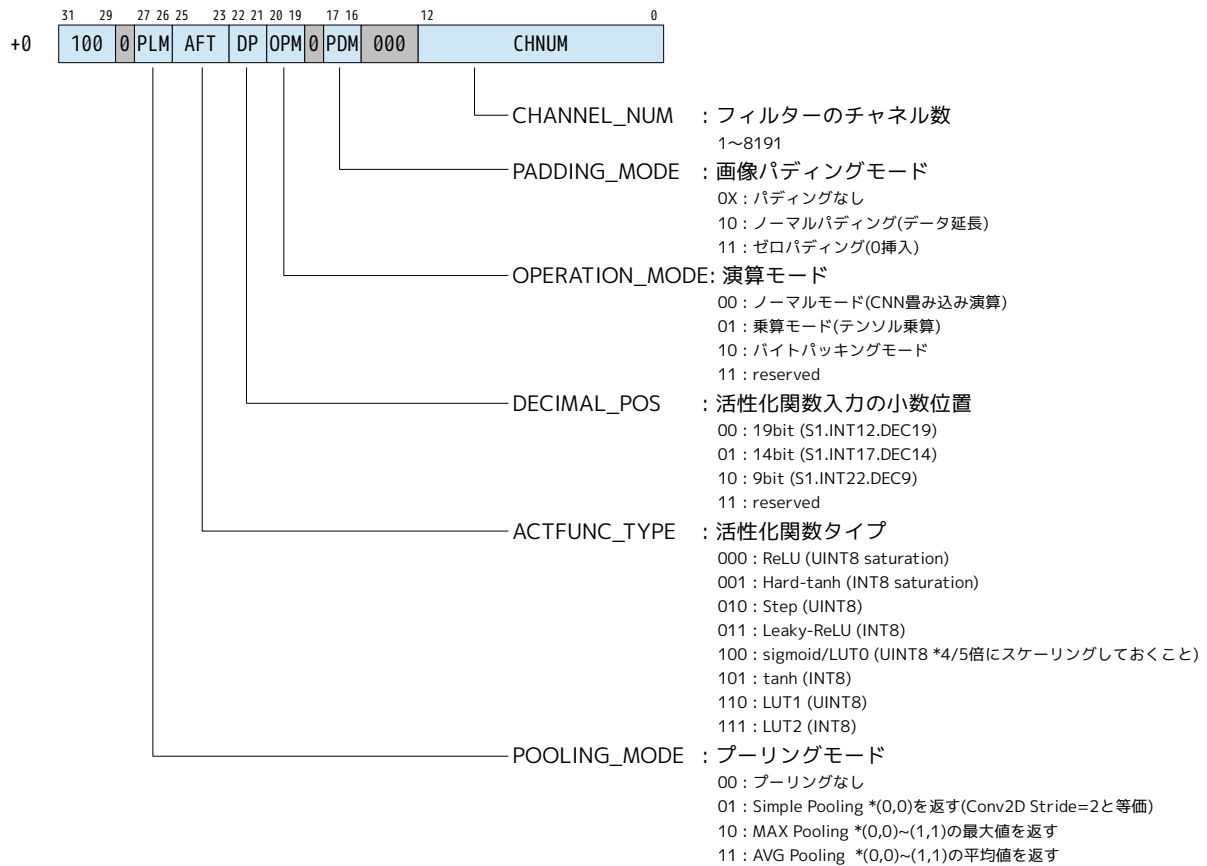
- [3:0] KERNEL0\_STATUS : カーネル0のステータス(RO)  
 bit0 : READY  
 bit1 : バスエラー発生(続行不能)  
 bit2 : 読み出しデータのアンダーフロー発生  
 bit3 : カーネル演算で飽和発生
- [7:4] KERNEL1\_STATUS : カーネル1のステータス(RO)  
 bit内容はカーネル0と同様
- [11:8] KERNEL2\_STATUS : カーネル2のステータス(RO)  
 bit内容はカーネル0と同様
- [15:12] KERNEL3\_STATUS : カーネル3のステータス(RO)  
 bit内容はカーネル0と同様
- [19:16] KERNEL4\_STATUS : カーネル4のステータス(RO)  
 bit内容はカーネル0と同様
- [23:20] KERNEL5\_STATUS : カーネル5のステータス(RO)  
 bit内容はカーネル0と同様
- [27:24] KERNEL6\_STATUS : カーネル6のステータス(RO)  
 bit内容はカーネル0と同様
- [31:28] KERNEL7\_STATUS : カーネル7のステータス(RO)  
 bit内容はカーネル0と同様



- [23:20] ACM\_STATUS : アキュムレーターのステータス(RO)  
 bit0 : READY  
 bit1 : バスエラー発生(続行不能)  
 bit2 : 内部バッファでアンダーフロー発生  
 bit3 : フィルター加算で飽和発生
- [27:24] FC\_STATUS : 全結合モジュールのステータス(RO)  
 bit0 : READY  
 bit1 : バスエラー発生(続行不能)  
 bit2 : reserved  
 bit3 : 全結合演算で飽和発生
- [28] WB\_STATUS : ライトバックモジュールのステータス(RO)  
 bit0 : READY

## デスクリプタ内容：フィルターパラメーター (1/2)

\* ONNXオペレーターのQLinearConv + MaxPool/AveragePoolに相当する機能



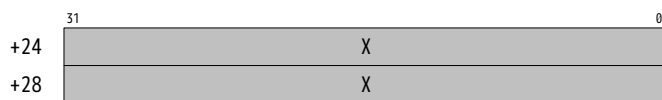
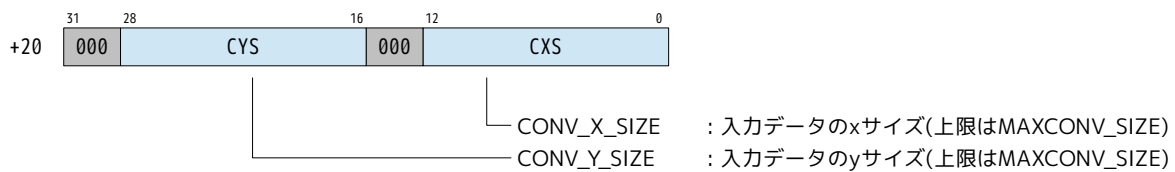
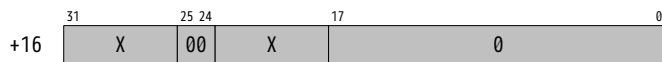
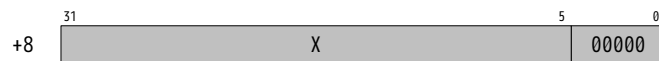
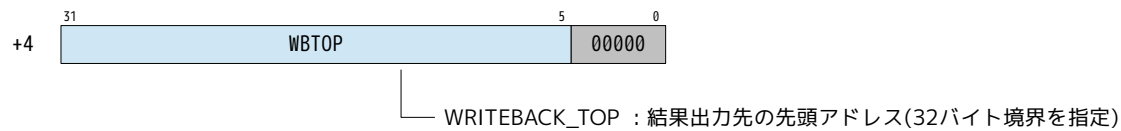
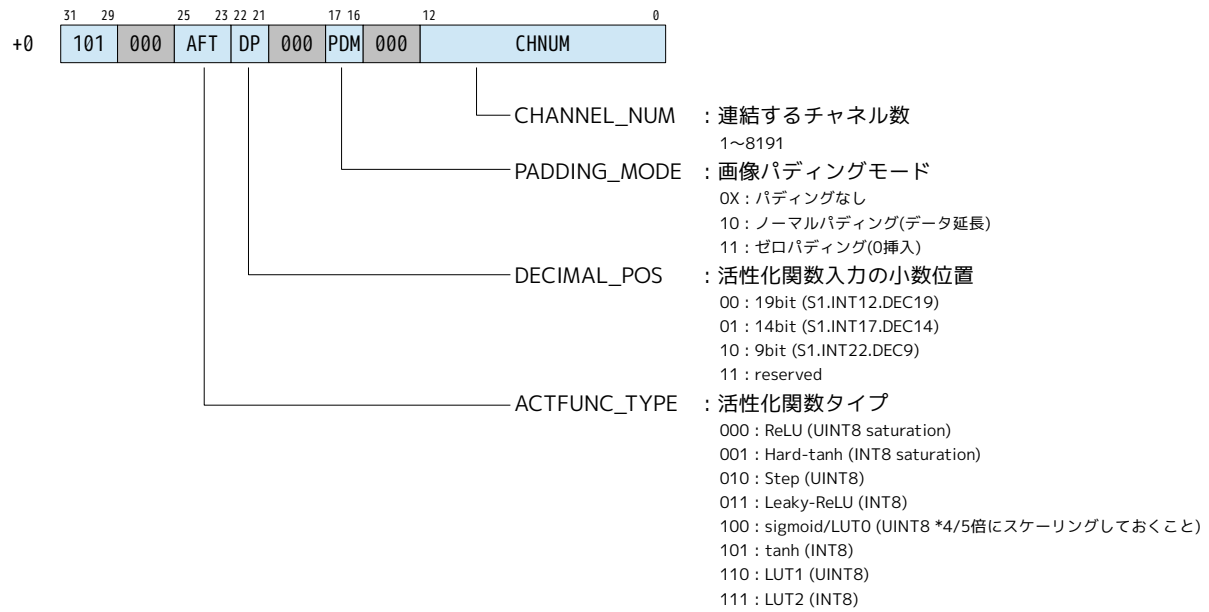


デスクリプタ内容：フィルターパラメーター (2/2)

	31	0
+24	X	
+28	X	

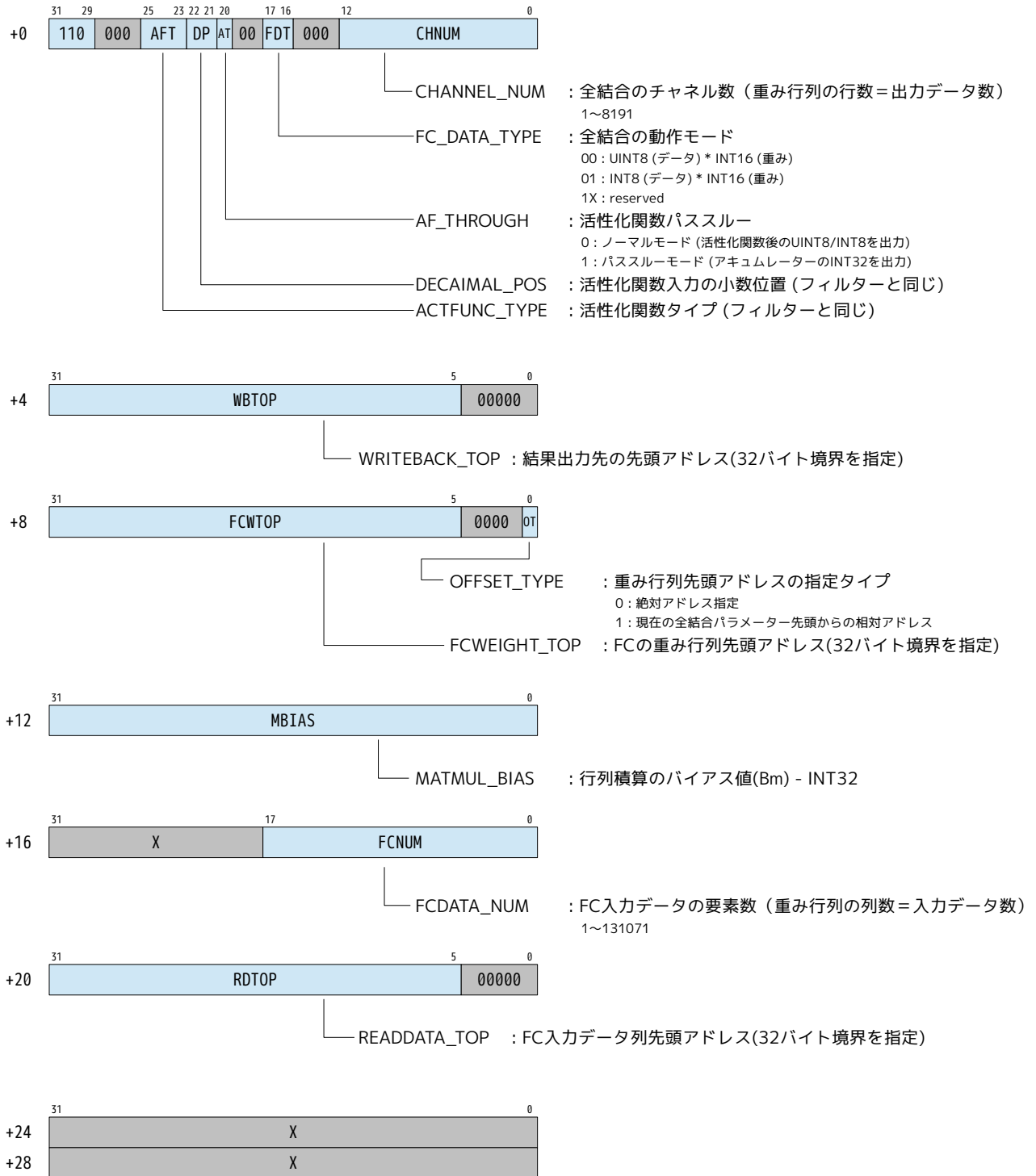
## デスクリプタ内容：シリアル化パラメーター

\* ONNXオペレーターのFlattenに相当する機能



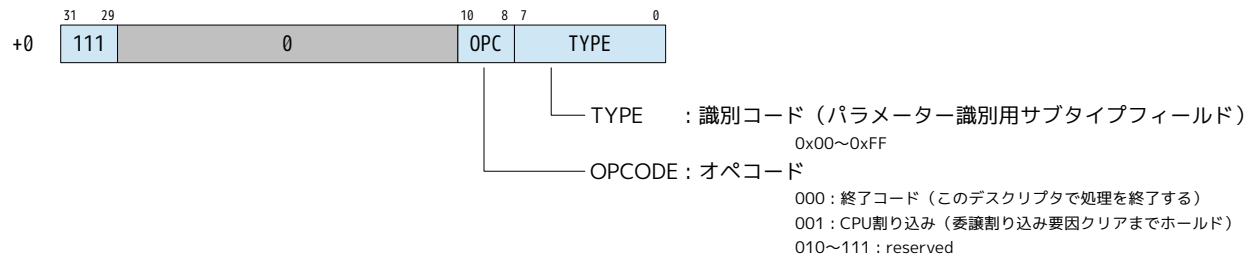
## デスクリプタ内容：全結合パラメーター

\* ONNXオペレーターのQLinearMatMulに相当する機能



## デスクリプタ内容：一時停止パラメーター

\* 対応していない機能をCPU側で処理させるためにデスクリプタ処理を一時停止または中断させる

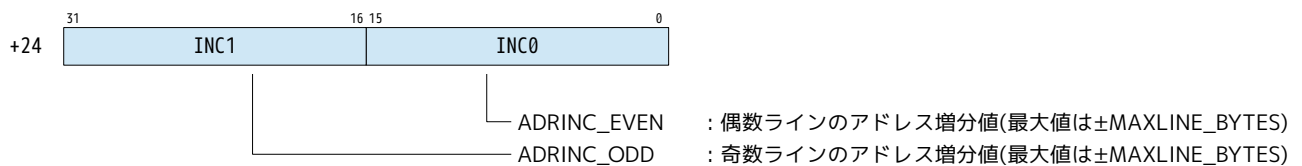
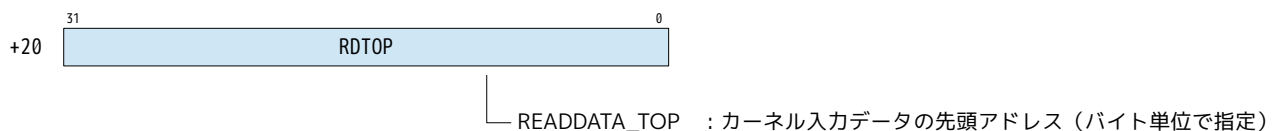
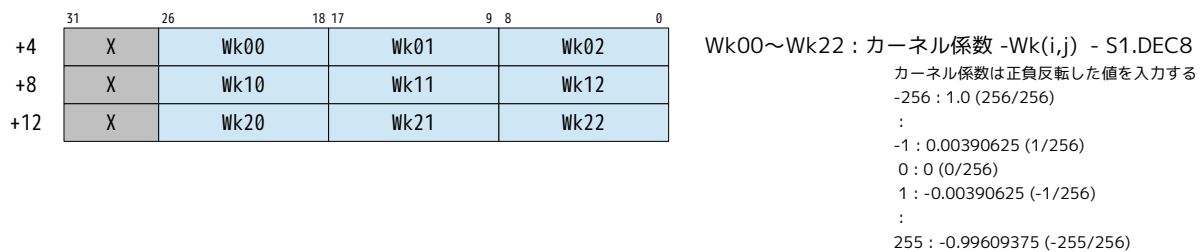
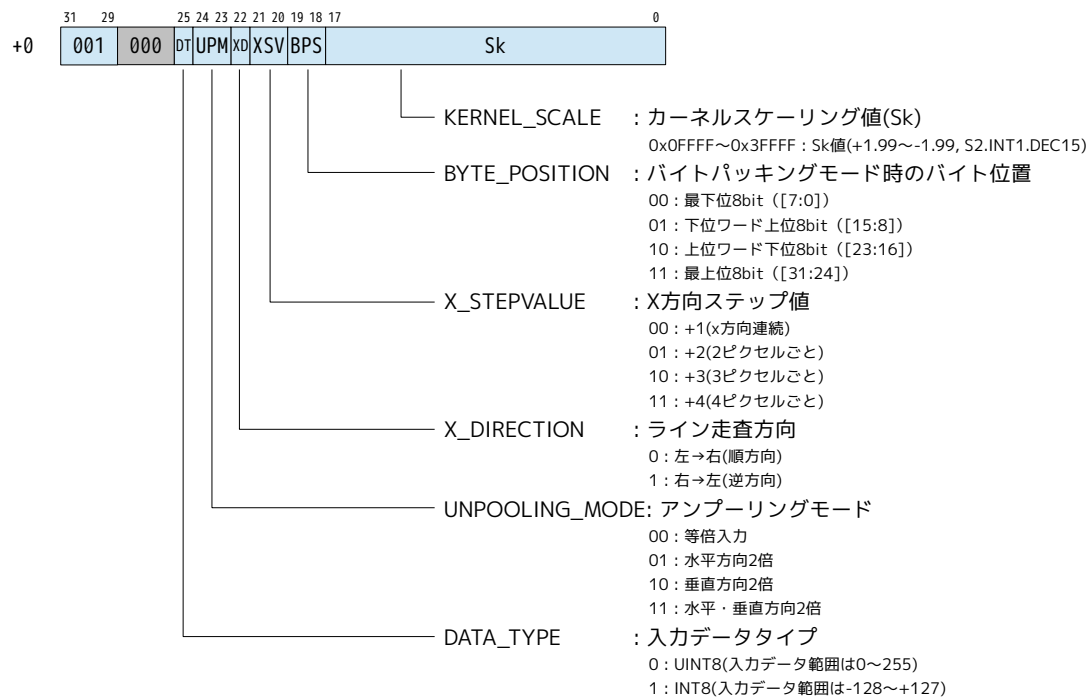


+4	CONTEXT1
+8	CONTEXT2
+12	CONTEXT3
+16	CONTEXT4
+20	CONTEXT5
+24	CONTEXT6
+28	CONTEXT7

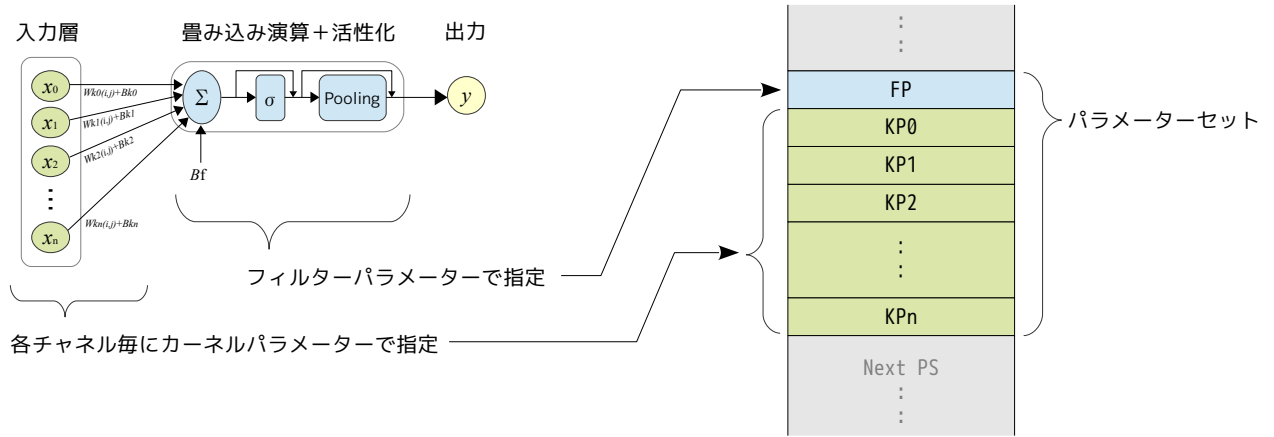
CONTEXTn : コンテキストデータ (OPCODE,TYPEにより内容は異なる)

## デスクリプタ内容：カーネルパラメーター

\* フィルター／シリアルライズパラメーターの後に記述するチャンネル毎のパラメーター



## デスクリプタ構造



カーネル出力結果はフィルター数分だけ加算される。

初期化時のバイアス(Bf)に加えて乱数項( $\epsilon$ )を加算することができる。ゲインはパラメーターで指定。

演算中にフィルター加算結果がオーバーフローした場合はOVFがセットされる（中断はしない）。

精度はカーネルのスケール係数(Sk)で任意に調整することができる。

ただし小数位置は活性化関数入力時点で以下の3種類に読み替えられる。

INT12:DEC19

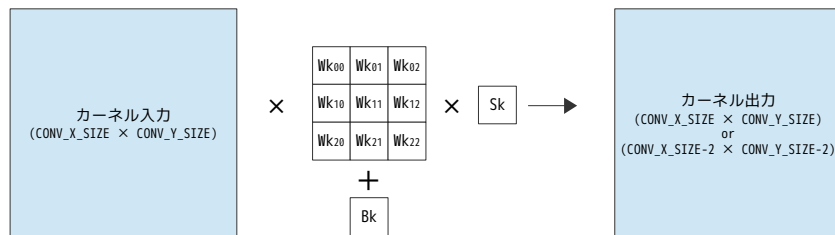
Ws= $\pm 1.0$ の範囲、フィルター数1(飽和なし保証)~8(カーネル係数総和1.0以下)

INT17:DEC14

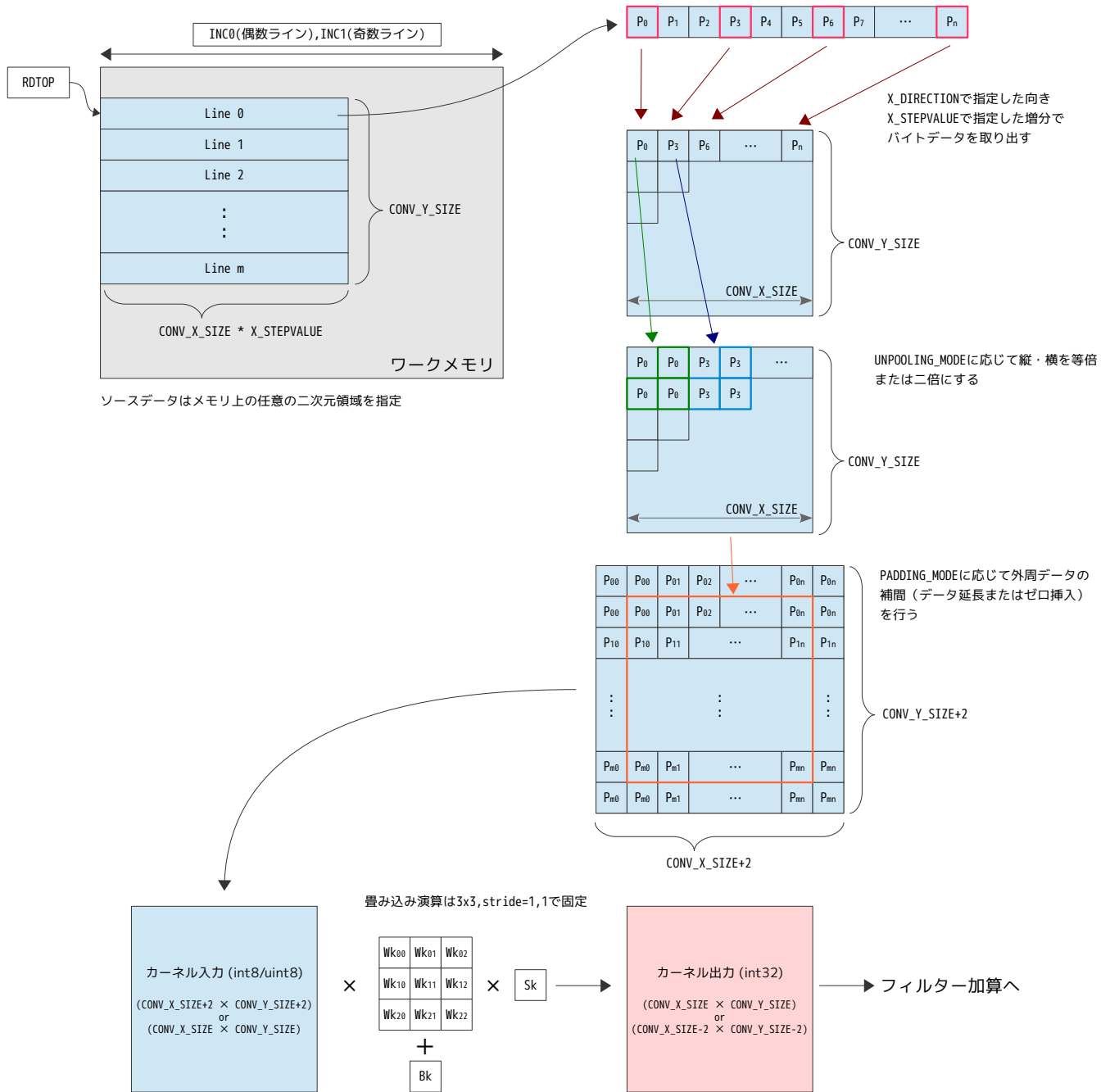
Ws= $\pm 1.0/32$ の範囲、フィルター数57(飽和なし保証)~256(カーネル係数総和1.0以下)

INT22:DEC9

Ws= $\pm 1.0/1024$ の範囲、フィルター数1834(飽和なし保証)~8192(カーネル係数総和1.0以下)



# 畳み込みカーネルの動作



# 活性化・プーリングの動作

