

SPU

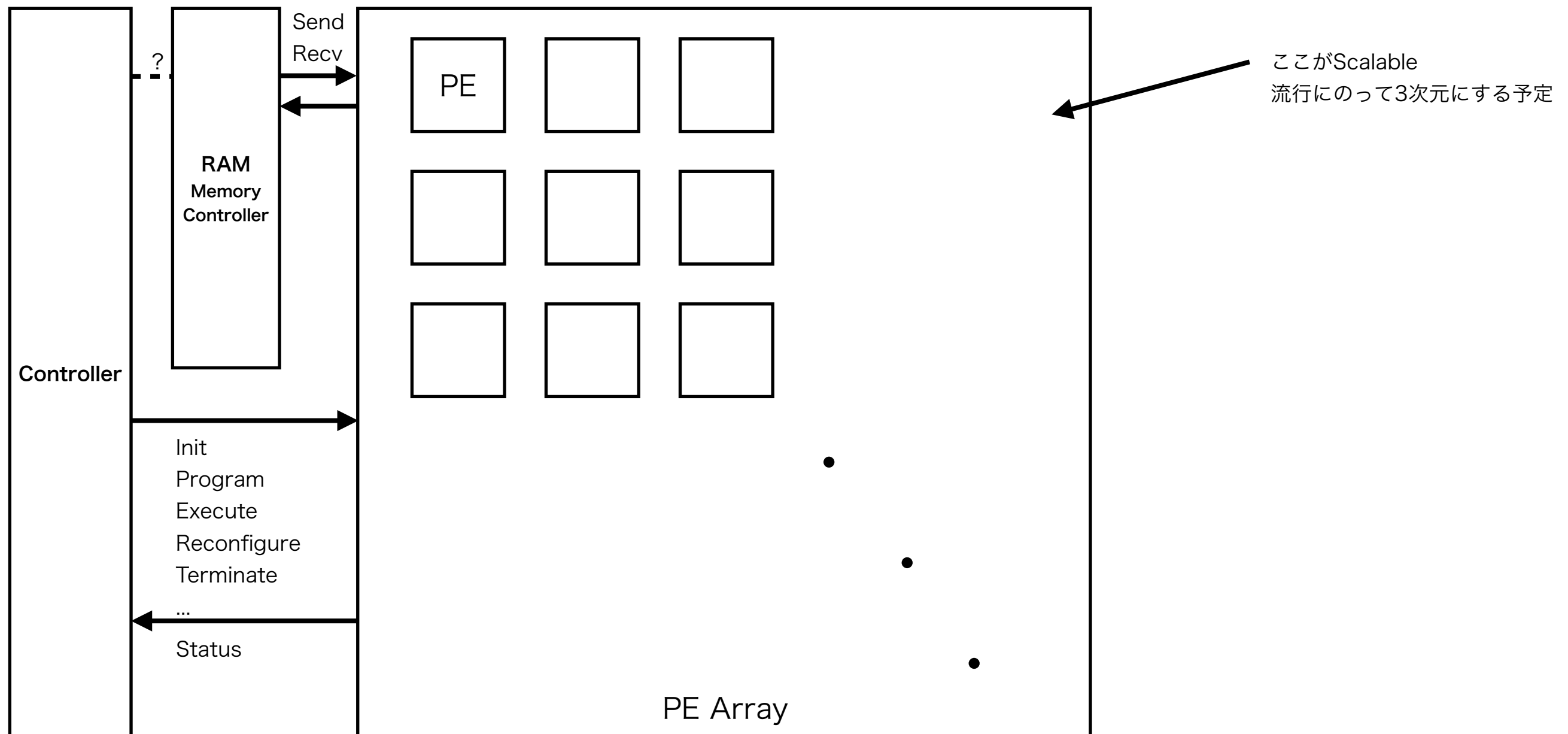
Ver0.0 2016/02/06

syskamk2

こんな感じ

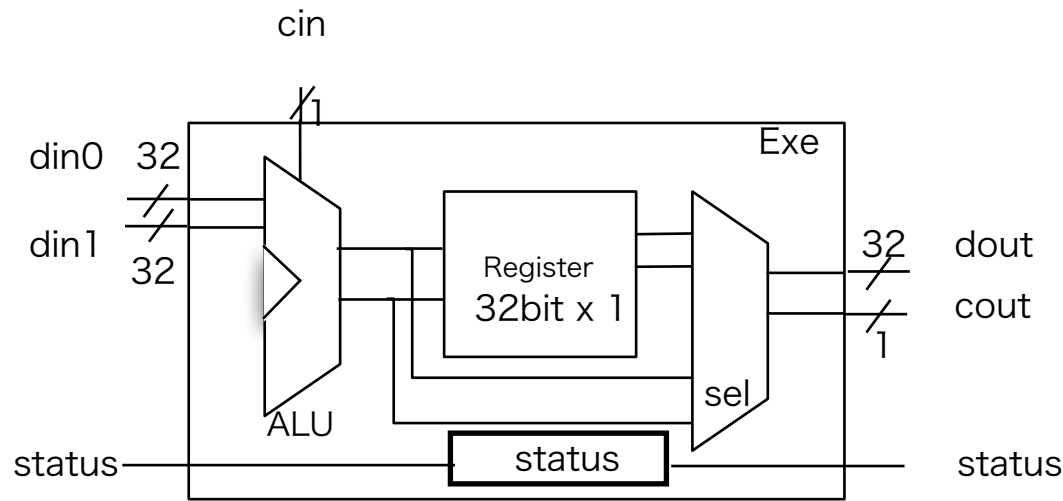
SPU (Scalable Processing Unit)

- ・ オーソドックスな粗粒度再構成可能HW
- ・ PE (Processing Element)が演算したりRAMとデータやりとりしたり



まずはエミュレータをScalaで書く。ハードウェア的な細かいところは今は考えない
全部1クロック、メモリ読み書き時間すら1クロック。

典型的なPE



コンフィギュレーションメモリ（とりあえず3枚持たせる）

status[0]がHの時だけ演算。status[1]がHで最後のデータ
statusは前段から伝播していく。*din0 din1用に2本必要かも

ALU機能

オペレータ	命令	input1	input2	input3	out1	out2
nop	0x0000	*	*	*	0	*
and	0x0001	din0	din1	*	din0&din1	*
or	0x0002	din0	din1	*	din0 din1	*
xor	0x0003	din0	din1	*	din0^din1	*
nand	0x0004	din0	din1	*	~(din0 & din1)	*
bitwise and	0x0010	din0	*	*	&din0	*
bitwise or	0x0020	din0	*	*	din0	*
not	0x0030	din0	*	*	~din0	*
sll	0x0100	din0	din1	*	din0 << din1	*
slr	0x0200	din0	din1	*	din0[31-din1:0] din0[31:31-din1-1] {din: > 0}	*
srl	0x0300	din0	din1	*	din0 >> din1	*
srr	0x0400	din0	din1	*	din0[din1-1:0] din0[31:din1] {din1: > 0}	*
srla	0x0500	din0	din1	*	din1 #din0[31] (din0 >> din1)[31-din1:0]	*
add	0x1000	din0	din1	*	din0 + din1	overflow
sub	0x2000	din0	din1	*	din0 - din1 - cin	underflow
mult	0x3000	din0	din1	*	din0[15:0] * din1[15:0]	*
div	0x4000	din0	din1	*	din0 / din1	div by 0
mod	0x5000	din0	din1	*	din0 % din1	div by 0

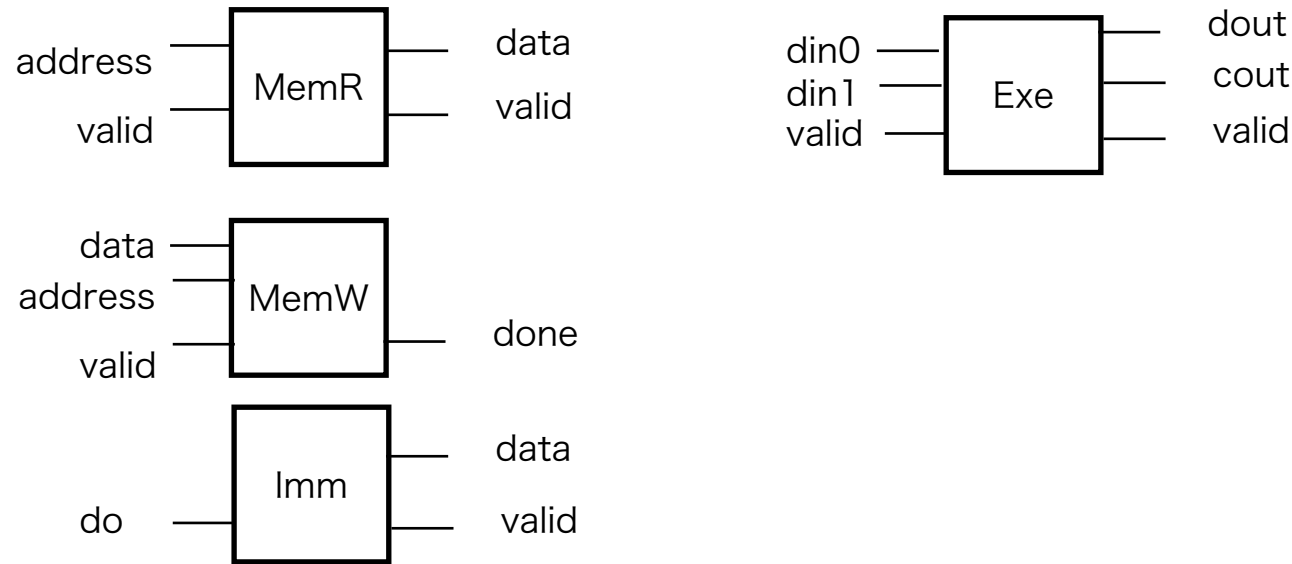
フラグ的に使う予定

ビット入れ替え、ビット選択、比較とか便利機能は後々追加

|| ビット連結
符号拡張

特殊なPE

メモリ読み込み用
メモリ書き込み用
即値生成（カウント機能等付き）



コンフィギュレーションメモリ



PE

種類	内容	
ALU命令	命令	32bit x 1
SEL	レジスタ/ALU出力	1bit x 1
接続情報	前段PE位置情報 (x, y, z)	24bit x 1

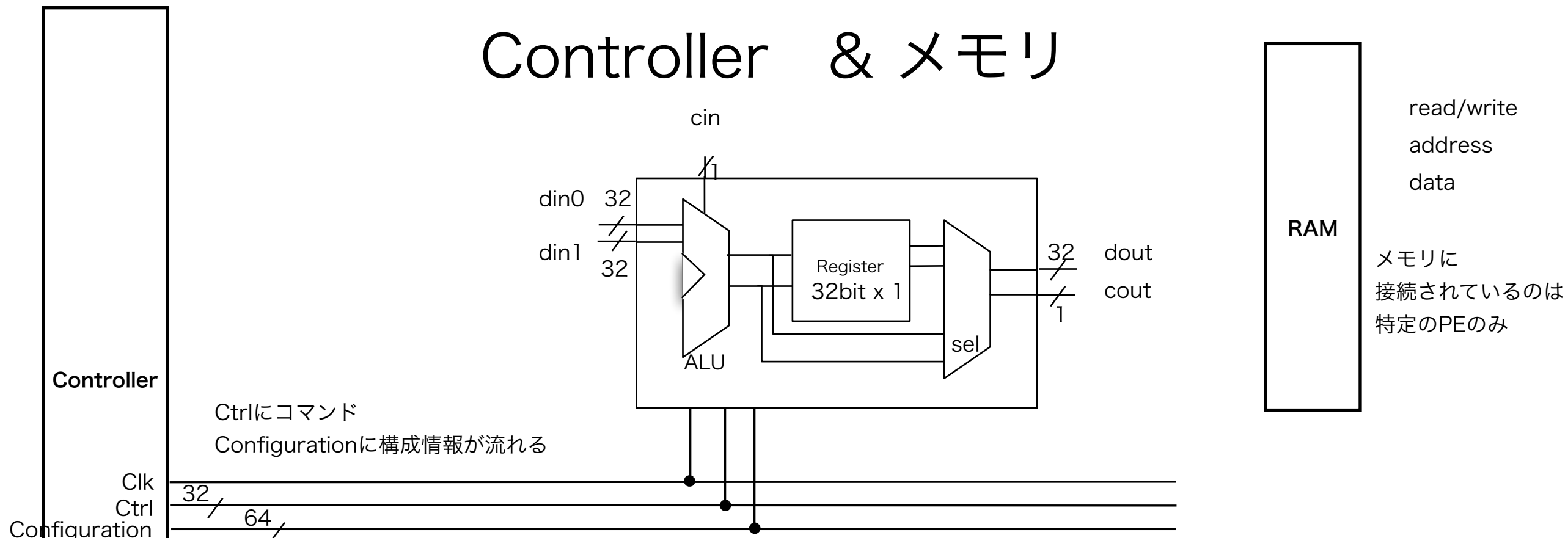
Imm

種類	内容	
初期値	初期値	32bit x 1
カウント/定数	カウンタになるか即値生成機になるか	1bit x 1
カウント増分値	カウント増分値	32bit x 1

MemR/MemW

種類	内容	
接続情報	前段PE位置情報 (x, y, z)	24bit x 1

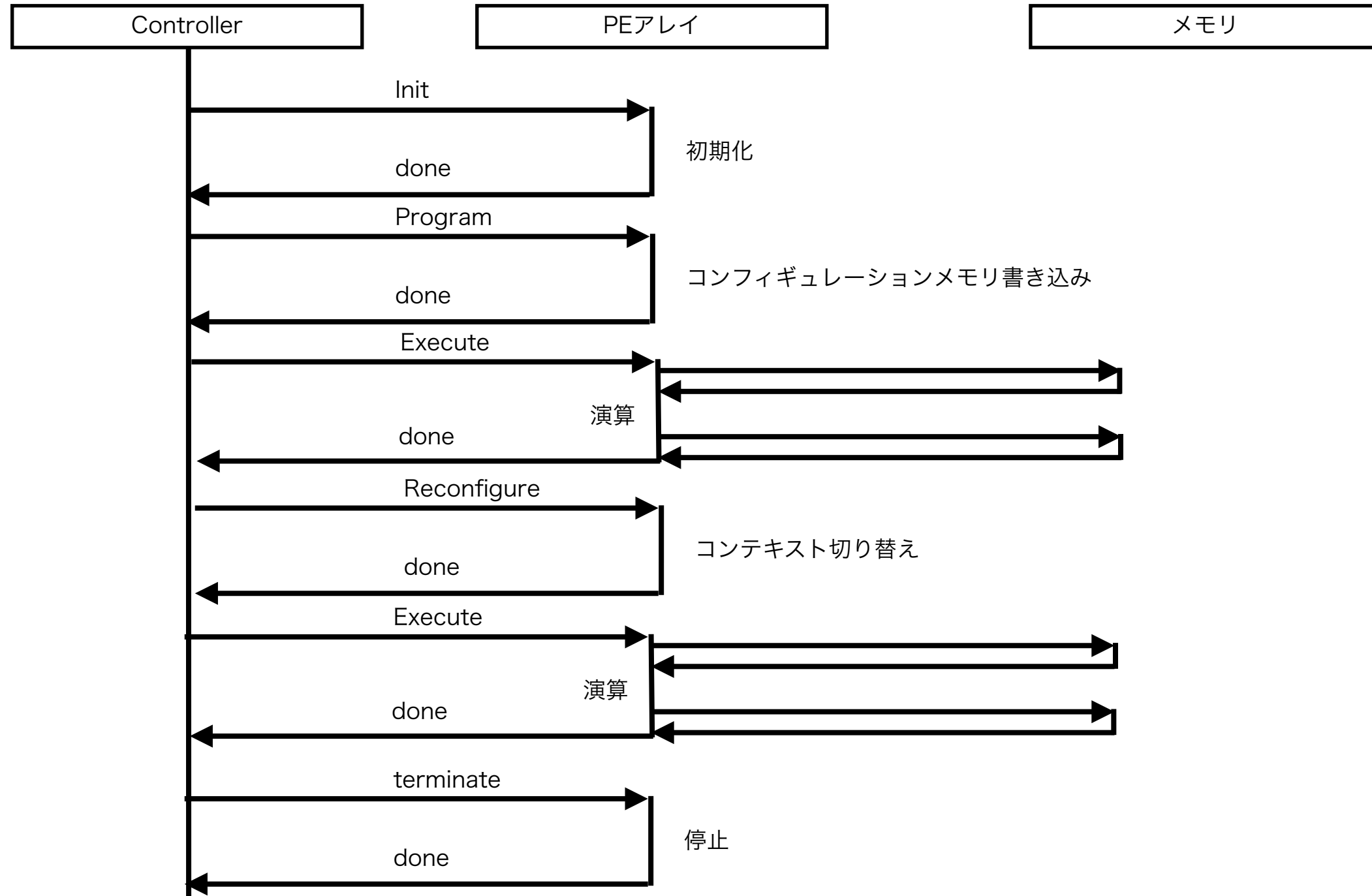
Controller & メモリ



機能	コマンド名	コマンド	引数	動作
制御無し	nop	0x00	-	何もしない
PEアレイ初期化	Init	0x01	-	レジスタ、コンフィギュレーションメモリリセット
構成情報転送	Program	0x02	構成情報 コンテキスト番号	コンフィギュレーションメモリ書き込み
演算開始	Execute	0x10	-	PEアレイがコンテキストメモリの内容を実行開始
PEアレイ再構成	Reconfigure	0x03	コンテキスト番号	PEの参照するコンテキスト切り替え
PEアレイ停止	Terminate	0x20	-	PEアレイ停止

構成情報
(ALU SEL 接続情報) を一セットでPE分送り込む

動作概略



雑なクラス設計

