

Proposed Instructions for the RISC-V Base P Extension  
Annex A  
Correspondences with Earlier P Extension Proposal

John Hauser

August 16, 2024

**Warning! This document is based on a draft proposal. The Base P extension that is eventually ratified by the RISC-V International Association is liable to differ from this proposal in many details.**

This document shows how the instructions from *Proposed Instructions for the RISC-V Base P Extension* correspond to instructions of the earlier P extension proposal.

The tables that follow are organized to list the new proposed instructions in close to the same order as the main document.

# 1 Instructions without multiplications

RV32/RV64		RV32/RV64	
New instruction	Earlier equivalent	New instruction	Earlier equivalent
PLI.B	—	PLI.H	—
PADD.B.B0	—	PLUI.H	—
PADD.B	ADD8	PADD.H.H0	—
PSUB.B	SUB8	PADD.H	ADD16
PSADD.B	KADD8	PSUB.H	SUB16
PSADDU.B	UKADD8	PSADD.H	KADD16
PSSUB.B	KSUB8	PSADDU.H	UKADD16
PSSUBU.B	UKSUB8	PSSUB.H	KSUB16
PAADD.B	RADD8	PSSUBU.H	UKSUB16
PAADDU.B	URADD8	PAADD.H	RADD16
PASUB.B	RSUB8	PAADDU.H	URADD16
PASUBU.B	URSUB8	PASUB.H	RSUB16
		PASUBU.H	URSUB16
		PSHIADD.H	—
		PSSH1SADD.H	—
		PAS.HX	CRAS16
		PSA.HX	CRSA16
		PSAS.HX	KCRAS16
		PSSA.HX	KCRSA16
		PAAS.HX	RCRAS16
		PASA.HX	RCRSA16
PDIF.B	—	PDIF.H	—
PDIFU.B	—	PDIFU.H	—
PSABS.B	KABS8	PSABS.H	KABS16
? PREDSUM.B	—	PREDSUM.H	—
? PREDSUMU.B	—	PREDSUMU.H	—
PDIFSUMU.B	PBSAD		
PDIFSUMAU.B	PBSADA		

RV32		RV64		RV64	
New instruction	Earlier equivalent	New instruction	Earlier equivalent	New instruction	Earlier equivalent
		PLI.W PLUL.W	— —		
SADD SADDU SSUB SSUBU AADD AADDU ASUB ASUBU	KADDW UKADDW KSUBW UKSUBW RADDW URADDW RSUBW URSUBW	PADD.W.W0 PADD.W PSUB.W PSADD.W PSADDU.W PSSUB.W PSSUBU.W PAADD.W PAADDU.W PASUB.W PASUBU.W	— ADD32 SUB32 KADD32 UKADD32 KSUB32 UKSUB32 RADD32 URADD32 RSUB32 URSUB32		
SH1ADD SSH1SADD	— —	PSH1ADD.W PSSH1SADD.W	— —	SH1ADD	—
		PAS.WX PSA.WX PSAS.WX PSSA.WX PAAS.WX PASA.WX	CRAS32 CRSA32 KCRAS32 KCRSA32 RCRAS32 RCRSA32		
		PREDSUM.W PREDSUMU.W	— —		

RV32/RV64		RV32/RV64	
New instruction	Earlier equivalent	New instruction	Earlier equivalent
		PSEXTB.H	SUNPKD820
		PSATI.H PUSATI.H	SCLIP16 UCLIP16
PSLLI.B PSLL.B.B0 PSRLI.B PSRL.B.B0 PSRAI.B PSRA.B.B0	SLLI8 SLL8 SRLI8 SRL8 SRAI8 SRA8	PSLLI.H PSLL.H.H0 PSRLI.H PSRL.H.H0 PSRAI.H PSRA.H.H0	SLLI16 SLL16 SRLI16 SRL16 SRAI16 SRA16
		PSSLAI.H PSRARI.H PSSHA.H.H0 PSSHAR.H.H0	KSLLI16 SRAI16.u — —
PMIN.B PMINU.B PMAX.B PMAXU.B	SMIN8 UMIN8 SMAX8 UMAX8	PMIN.H PMINU.H PMAX.H PMAXU.H	SMIN16 UMIN16 SMAX16 UMAX16
PMSEQ.B PMSLT.B PMSLTU.B	CMPEQ8 SCMPLT8 UCMPLT8	PMSEQ.H PMSLT.H PMSLTU.H	CMPEQ16 SCMPLT16 UCMPLT16

RV32		RV64		RV64	
New instruction	Earlier equivalent	New instruction	Earlier equivalent	New instruction	Earlier equivalent
SATI USATI	SCLIP32 UCLIP32	PSEXTB.W PSEXTH.W PSATI.W PUSATI.W	— — SCLIP32 UCLIP32	SATI USATI	— —
		PSLLI.W PSLL.W.W0 PSRLI.W PSRL.W.W0 PSRAI.W PSRA.W.W0	SLLI32 SLL32 SRLI32 SRL32 SRAI32 SRA32		
SSLAI SRARI SSHA SSHAR	KSLLIW SRAI.u — —	PSSLAI.W PSRARI.W PSSHA.W.W0 PSSHAR.W.W0	KSLLI32 SRAI32.u — —	SRARI SHA SHAR	SRAI.u — —
MIN MINU MAX MAXU	MIN MINU MAX MAXU	PMIN.W PMINU.W PMAX.W PMAXU.W	SMIN32 UMIN32 SMAX32 UMAX32	MIN MINU MAX MAXU	MIN MINU MAX MAXU
? MSEQ ? MSLT ? MSLTU	— — —	PMSEQ.W PMSLT.W PMSLTU.W	— — —		

RV32/RV64	
New instruction	Earlier equivalent
PPACK.H	—
PPACKBT.H	—
PPACKTB.H	—
PPACKT.H	—
REV8	—

RV32	
New instruction	Earlier equivalent
PACK	PACK
PACKBT	PKTB16 (*1)
PACKTB	PKBT16 (*1)
PACKT	PACKU

RV64	
New instruction	Earlier equivalent
PPACK.W	PKBB16 (*1)
PPACKBT.W	PKTB16 (*1)
PPACKTB.W	PKBT16 (*1)
PPACKT.W	PKTT16 (*1)
REV16	—

RV64	
New instruction	Earlier equivalent
PACK	PACK
PACKBT	PKTB32 (*1)
PACKTB	PKBT32 (*1)
PACKT	PACKU

(\*1) Swap the source operands.

RV64	
New instruction	Earlier equivalent
ZIP8P	—
ZIP8HP	—
UNZIP8P	—
UNZIP8HP	—

RV64	
New instruction	Earlier equivalent
ZIP16P	—
ZIP16HP	—
UNZIP16P	—
UNZIP16HP	—

RV32/RV64	
New instruction	Earlier equivalent
ABS	—
CLZ	CLZ
CLS	—
REV	REV
? SLXI	—
? SLX	—
? SRXI	≈ FSRI (*1)
SRX	—
MVM	≈ CMIX (*2)
MVMN	≈ CMIX (*3)
MERGE	≈ CMIX (*4)

RV64	
New instruction	Earlier equivalent
ABSW	≈ KABSW (*5)
CLZW	—
CLSW	—

(\*1) The earlier FSRI exists only for RV32. With an immediate operand  $i$  in the range 0–31 inclusive, RV32's new SRXI  $rd,rs1,i = FSRI\ rd,rd,rs1,i$ .

(\*2) New MVM  $rd,rs1,rs2 =$  earlier CMIX  $rd,rs1,rs2,rd$ .

(\*3) New MVMN  $rd,rs1,rs2 =$  earlier CMIX  $rd,rd,rs2,rs1$ .

(\*4) New MERGE  $rd,rs1,rs2 =$  earlier CMIX  $rd,rs2,rd,rs1$ .

(\*5) The new ABSW gives an unsigned result, while the earlier KABSW delivers a signed result, with saturation.

RV32, register-pair destination			Earlier equivalent
New instruction			
PWADD.B	PWADD.H	WADD	<i>none</i>
PWADDA.B	PWADDA.H	WADDA	
PWADDU.B	PWADDU.H	WADDU	
PWADDAU.B	PWADDAU.H	WADDAU	
PWSUB.B	PWSUB.H	WSUB	
PWSUBA.B	PWSUBA.H	WSUBA	
PWSUBU.B	PWSUBU.H	WSUBU	
PWSUBAU.B	PWSUBAU.H	WSUBAU	
PWSLLI.B	PWSLLI.H	WSLLI	
PWSLL.B.B0	PWSLL.H.H0	WSLL	
PWSLAI.B	PWSLAI.H	WSLAI	
PWSLA.B.B0	PWSLA.H.H0	WSLA	
WZIP8P	WZIP16P		<i>none</i>

RV32, register-pair operands			Earlier equivalent
New instruction			
PLI.DB	PLI.DH		<i>none</i>
	PLUI.DH		
PADD.DB	PADD.DH	PADD.DW	
PSUB.DB	PSUB.DH	PSUB.DW	
PSADD.DB	PSADD.DH	PSADD.DW	
PSADDU.DB	PSADDU.DH	PSADDU.DW	
PSSUB.DB	PSSUB.DH	PSSUB.DW	
PSSUBU.DB	PSSUBU.DH	PSSUBU.DW	
PAADD.DB	PAADD.DH	PAADD.DW	
PAADDU.DB	PAADDU.DH	PAADDU.DW	
PASUB.DB	PASUB.DH	PASUB.DW	
PASUBU.DB	PASUBU.DH	PASUBU.DW	
	PSH1ADD.DH	PSH1ADD.DW	
	PSSH1SADD.DH	PSSH1SADD.DW	
	PAS.DHX		
	PSA.DHX		
	PSAS.DHX		
	PSSA.DHX		
	PAAS.DHX		
	PASA.DHX		
PDIF.DB	PDIF.DH		
PDIFU.DB	PDIFU.DH		
PSABS.DB	PSABS.DH		

RV32, register-pair operands	
New instruction	Earlier equivalent
ADDD	ADD64
SUBD	SUB64

RV32, register-pair first source (only)		
New instruction		Earlier equivalent
? PREDSUM.DB	PREDSUM.DH	<i>none</i>
? PREDSUMU.DB	PREDSUMU.DH	

RV32, register-pair operands			Earlier equivalent
New instruction			
	PSEXTB.DH	PSEXTB.DW	<i>none</i>
	PSATH.DH	PSATH.DW	
	PUSATH.DH	PUSATH.DW	
PSLLI.DB	PSLLI.DH	PSLLI.DW	
PSRLI.DB	PSRLI.DH	PSRLI.DW	
PSRAI.DB	PSRAI.DH	PSRAI.DW	
	PSSLAI.DH	PSSLAI.DW	
	PSRARI.DH	PSRARI.DW	
PMIN.DB	PMIN.DH	PMIN.DW	
PMINU.DB	PMINU.DH	PMINU.DW	
PMAX.DB	PMAX.DH	PMAX.DW	
PMAXU.DB	PMAXU.DH	PMAXU.DW	
PMSEQ.DB	PMSEQ.DH	? PMSEQ.DW	
PMSLT.DB	PMSLT.DH	? PMSLT.DW	
PMSLTU.DB	PMSLTU.DH	? PMSLTU.DW	

RV32, register-pair first source and destination			Earlier equivalent
New instruction			
PADD.DB.B0	PADD.DH.H0	PADD.DW.W	<i>none</i>
PSLL.DB.B0	PSLL.DH.H0	PSLL.DW.W	
PSRL.DB.B0	PSRL.DH.H0	PSRL.DW.W	
PSRA.DB.B0	PSRA.DH.H0	PSRA.DW.W	
	PSSHA.DH.H0	PSSHA.DW.W	
	PSSHAR.DH.H0	PSSHAR.DW.W	

RV32, register-pair operands		Earlier equivalent
New instruction		
PPACK.DH	PPACK.DW	<i>none</i>
PPACKBT.DH	PPACKBT.DW	
PPACKTB.DH	PPACKTB.DW	
PPACKT.DH	PPACKT.DW	

RV32, register-pair first source (only)			Earlier equivalent
New instruction			
PNSRLI.B	PNSRLI.H	NSRLI	<i>none</i>
PNSRL.B.B0	PNSRL.H.H0	NSRL	
PNSRAI.B	PNSRAI.H	NSRAI	
PNSRA.B.B0	PNSRA.H.H0	NSRA	
PNSRARI.B	PNSRARI.H	NSRARI	
PNSRAR.B.B0	PNSRAR.H.H0	NSRAR	
PNCLIP.I.B	PNCLIP.I.H	NCLIP.I	
PNCLIP.B.B0	PNCLIP.H.H0	NCLIP	
PNCLIPRI.B	PNCLIPRI.H	NCLIPRI	
PNCLIPR.B.B0	PNCLIPR.H.H0	NCLIPR	
PNCLIPIU.B	PNCLIPIU.H	NCLIPIU	
PNCLIPU.B.B0	PNCLIPU.H.H0	NCLIPU	
PNCLIPRIU.B	PNCLIPRIU.H	NCLIPRIU	
PNCLIPRU.B.B0	PNCLIPRU.H.H0	NCLIPRU	



## 2 Instructions that perform multiplications

RV32/RV64	
New instruction	Earlier equivalent
PMULH.H	—
PMULHR.H	—
PMULHSU.H	—
PMULHRSU.H	—
PMULHU.H	—
PMULHRU.H	—
PMULQ.H	KHM16
PMULQR.H	—
PMHACC.H	—
PMHRACC.H	—
PMHACCSU.H	—
PMHRACCSU.H	—
PMHACCU.H	—
PMHRACCU.H	—

RV32	
New instruction	Earlier equivalent
MULHR	SMMUL.u
MULHRSU	—
MULHRU	—
MULQ	KWMMUL
MULQR	KWMMUL.u
MHACC	≈ KMMAC (*1)
MHRACC	≈ KMMAC.u (*1)
MHACCSU	—
MHRACCSU	—
MHACCU	—
MHRACCU	—

RV64	
New instruction	Earlier equivalent
PMULH.W	SMMUL
PMULHR.W	SMMUL.u
PMULHSU.W	—
PMULHRSU.W	—
PMULHU.W	—
PMULHRU.W	—
PMULQ.W	KWMMUL
PMULQR.W	KWMMUL.u
PMHACC.W	≈ KMMAC (*1)
PMHRACC.W	≈ KMMAC.u (*1)
PMHACCSU.W	—
PMHRACCSU.W	—
PMHACCU.W	—
PMHRACCU.W	—

(\*1) The new instruction does not saturate the addition, while the earlier instruction does.

RV32	
New instruction	Earlier equivalent
MQACC.H $nn$	—
MQRACC.H $nn$	—

RV64	
New instruction	Earlier equivalent
PMQACC.W.H $pp$	—
PMQRACC.W.H $pp$	—

RV32/RV64	
New instruction	Earlier equivalent
PMQ2ADD.H	—
PMQ2ADDA.H	—
PMQR2ADD.H	—
PMQR2ADDA.H	—

RV64	
New instruction	Earlier equivalent
MQACC.W $nn$	—
MQRACC.W $nn$	—
PMQ2ADD.W	—
PMQ2ADDA.W	—
PMQR2ADD.W	—
PMQR2ADDA.W	—

RV32/RV64	
New instruction	Earlier equivalent
PMUL.H.B $pp$	—
PMULSU.H.B $pp$	—
PMULU.H.B $pp$	—

RV64	
New instruction	Earlier equivalent
PMUL.W.H $pp$	SM $pp$ 16 (*2)
PMULSU.W.H $pp$	—
PMULU.W.H $pp$	—
PMACC.W.H $pp$	$\approx$ KMA $pp$ (*2, 4)
PMACCSU.W.H $pp$	—
PMACCU.W.H $pp$	—

RV32	
New instruction	Earlier equivalent
MUL.H $nn$	SM $pp$ 16 (*1)
MULSU.H $nn$	—
MULU.H $nn$	—
MACC.H $nn$	$\approx$ KMA $pp$ (*1, 4)
MACCSU.H $nn$	—
MACCU.H $nn$	—

RV64	
New instruction	Earlier equivalent
MUL.W $nn$	SM $pp$ 32 (*1)
MULSU.W $nn$	—
MULU.W $nn$	$\approx$ MULR64 (*5)
MACC.W $nn$	$\approx$ KMA $pp$ 32 (*1, 4)
MACCSU.W $nn$	—
MACCU.W $nn$	—

RV32/RV64	
New instruction	Earlier equivalent
PM2ADD.H	$\approx$ KMDA (*4)
PM2ADDA.H	$\approx$ KMADA (*4)
PM2ADDSU.H	—
PM2ADDASU.H	—
PM2ADDU.H	—
PM2ADDAU.H	—
PM2ADD.HX	$\approx$ KMXDA (*4)
PM2ADDA.HX	$\approx$ KMAXDA (*4)
PM2SADD.H	KMDA
PM2SADD.HX	KMXDA
PM2SUB.H	SMDRS
PM2SUBA.H	$\approx$ KMADRS (*4)
PM2SUB.HX	SMXDS (*3)
PM2SUBA.HX	$\approx$ KMAXDS (*3, 4)

PM2ADD.W	$\approx$ KMDA32 (*4)
PM2ADDA.W	SMAR64
PM2ADDSU.W	—
PM2ADDASU.W	—
PM2ADDU.W	—
PM2ADDAU.W	UMAR64
PM2ADD.WX	$\approx$ KMXDA32 (*4)
PM2ADDA.WX	$\approx$ KMAXDA32 (*4)
PM2SUB.W	SMDRS32
PM2SUBA.W	$\approx$ KMADRS32 (*4)
PM2SUB.WX	SMXDS32 (*3)
PM2SUBA.WX	$\approx$ KMAXDS32 (*3, 4)

- (\*1) For the elements to select from each source operand, the earlier instruction has  $p$  being ‘B’ or ‘T’, corresponding to an  $n$  of ‘0’ or ‘1’.
- (\*2) For the elements to select from each source operand, specified by  $p$ , the earlier instruction has ‘B’ or ‘T’ instead of ‘E’ or ‘O’.
- (\*3) Swap the source operands.
- (\*4) The new instruction does not saturate the addition(s), while the earlier instruction does.
- (\*5) The earlier MULR64 is the same as the new MULU.W00 only.

RV32/RV64		RV64	
New instruction	Earlier equivalent	New instruction	Earlier equivalent
PM4ADD.B	—	PM4ADD.H	—
PM4ADDA.B	SMAQA	PM4ADDA.H	SMALDA
PM4ADDSU.B	—	PM4ADDSU.H	—
PM4ADDASU.B	SMAQA.SU	PM4ADDASU.H	—
PM4ADDU.B	—	PM4ADDU.H	—
PM4ADDAU.B	UMAQA	PM4ADDAU.H	—

RV32/RV64	
New instruction	Earlier equivalent
PMULH.H.B $p$	—
PMULHSU.H.B $p$	—
PMHACC.H.B $p$	—
PMHACCSU.H.B $p$	—

RV32		RV64	
New instruction	Earlier equivalent	New instruction	Earlier equivalent
MULH.H $n$	SMMW $p$ (*1)	PMULH.W.H $p$	SMMW $p$ (*2)
MULHSU.H $n$	—	PMULHSU.W.H $p$	—
MHACC.H $n$	$\approx$ KMMAW $p$ (*1, 3)	PMHACC.W.H $p$	$\approx$ KMMAW $p$ (*2, 3)
MHACCSU.H $n$	—	PMHACCSU.W.H $p$	—

- (\*1) For the element to select from the second operand, the earlier instruction has  $p$  being ‘B’ or ‘T’, corresponding to an  $n$  of ‘0’ or ‘1’.
- (\*2) For the elements to select from the second operand, specified by  $p$ , the earlier instruction has ‘B’ or ‘T’ instead of ‘E’ or ‘O’.
- (\*3) The new instruction does not saturate the addition, while the earlier instruction does.

RV32 register-pair destination	
New instruction	Earlier equivalent
PWMUL.B	SMUL8
PWMULSU.B	—
PWMULU.B	UMUL8

RV32 register-pair destination	
New instruction	Earlier equivalent
PMQWACC.H	—
PMQRWACC.H	—
PWMUL.H	SMUL16
PWMULSU.H	—
PWMULU.H	UMUL16
PWMACC.H	—
PWMACCSU.H	—
PWMACCU.H	—
PM2WADD.H	—
PM2WADDA.H	SMALDA
PM2WADDSU.H	—
PM2WADDASU.H	—
PM2WADDU.H	—
PM2WADDAU.H	—
PM2WADD.HX	—
PM2WADDA.HX	SMALXDA
PM2WSUB.H	—
PM2WSUBA.H	SMALDRS
PM2WSUB.HX	—
PM2WSUBA.HX	SMALXDS (*1)

RV32 register-pair destination	
New instruction	Earlier equivalent
MQWACC	—
MQRWACC	—
WMUL	MULSR64
WMULSU	—
WMULU	MULR64
WMACC	SMAR64
WMACCSU	—
WMACCU	UMAR64

(\*1) Swap the source operands.