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

John Hauser

December 18, 2024

Warning! This document is based on a draft proposal and is not an official document of the RISC-V International Association. The Base P extension that is eventually ratified by RISC-V International is liable to differ from this document in many details.

In this document, I show how the instructions from my *Proposed Instructions for the RISC-V Base P Extension* correspond to instructions of the earlier P extension proposal.

The changes in this version (012) from the previous one (011) are explained here:
`https://lists.riscv.org/g/tech-p-ext/message/511`

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 New instruction	Earlier equivalent
PLI.B	—
PADD.B.B0	—
PADD.B	ADD8
PSUB.B	SUB8
PSADD.B	KADD8
PSADDU.B	UKADD8
PSSUB.B	KSUB8
PSSUBU.B	UKSUB8
PAADD.B	RADD8
PAADDU.B	URADD8
PASUB.B	RSUB8
PASUBU.B	URSUB8
PDIF.B	—
PDIFU.B	—
PSABS.B	KABS8
PREDSUM.B	—
PREDSUMU.B	—
PDIFSUMU.B	PBSAD
PDIFSUMAU.B	PBSADA

RV32/RV64 New instruction	Earlier equivalent
PLI.H	—
PLUI.H	—
PADD.H.H0	—
PADD.H	ADD16
PSUB.H	SUB16
PSADD.H	KADD16
PSADDU.H	UKADD16
PSSUB.H	KSUB16
PSSUBU.H	UKSUB16
PAADD.H	RADD16
PAADDU.H	URADD16
PASUB.H	RSUB16
PASUBU.H	URSUB16
PSH1ADD.H	—
PSSH1SADD.H	—
PAS.HX	CRAS16
PSA.HX	CRSA16
PSAS.HX	KCRAS16
PSSA.HX	KCRSA16
PAAS.HX	RCRAS16
PASA.HX	RCRSA16
PDIF.H	—
PDIFU.H	—
PSABS.H	KABS16
PREDSUM.H	—
PREDSUMU.H	—

RV32	
New instruction	Earlier equivalent
SADD	KADDW
SADDU	UKADDW
SSUB	KSUBW
SSUBU	UKSUBW
AADD	RADDW
AADDU	URADDW
ASUB	RSUBW
ASUBU	URSUBW
SH1ADD	—
SSH1SADD	—

RV64	
New instruction	Earlier equivalent
PLI.W	—
PLUI.W	—
PADD.W.W0	—
PADD.W	ADD32
PSUB.W	SUB32
PSADD.W	KADD32
PSADDU.W	UKADD32
PSSUB.W	KSUB32
PSSUBU.W	UKSUB32
PAADD.W	RADD32
PAADDU.W	URADD32
PASUB.W	RSUB32
PASUBU.W	URSUB32
PSH1ADD.W	—
PSSH1SADD.W	—
PAS.WX	CRAS32
PSA.WX	CRSA32
PSAS.WX	KCRAS32
PSSA.WX	KCRSA32
PAAS.WX	RCRAS32
PASA.WX	RCRSA32
PREDSUM.W	—
PREDSUMU.W	—

RV64	
New instruction	Earlier equivalent
SH1ADD	—

RV32/RV64	
New instruction	Earlier equivalent
PSLLI.B	SLLI8
PSLL.B.B0	—
PSRLI.B	SRLI8
PSRL.B.B0	—
PSRAI.B	SRAI8
PSRA.B.B0	—
PMIN.B	SMIN8
PMINU.B	UMIN8
PMAX.B	SMAX8
PMAXU.B	UMAX8
PMSEQ.B	CMPEQ8
PMSLT.B	SCMPLT8
PMSLTU.B	UCMPLT8

RV32/RV64	
New instruction	Earlier equivalent
PSEXTB.H	SUNPKD820
PSATI.H	SCLIP16
PUSATI.H	UCLIP16
PSLLI.H	SLLI16
PSLL.H.H0	—
PSRLI.H	SRLI16
PSRL.H.H0	—
PSRAI.H	SRAI16
PSRA.H.H0	—
PSSLAI.H	KSLLI16
PSRARI.H	SRAI16.u
PSSHA.H.H0	—
PSSHAR.H.H0	—
PMIN.H	SMIN16
PMINU.H	UMIN16
PMAX.H	SMAX16
PMAXU.H	UMAX16
PMSEQ.H	CMPEQ16
PMSLT.H	SCMPLT16
PMSLTU.H	UCMPLT16

RV32	
New instruction	Earlier equivalent
SATI	SCLIP32
USATI	UCLIP32
SSLAI	KSLLIW
SRARI	SRAI.u
SSHA	—
SSHAR	—
MIN	MIN
MINU	MINU
MAX	MAX
MAXU	MAXU
MSEQ	—
MSLT	—
MSLTU	—

RV64	
New instruction	Earlier equivalent
PSEXTB.W	—
PSEXTH.W	—
PSATI.W	SCLIP32
PUSATI.W	UCLIP32
PSLLI.W	SLLI32
PSLL.W.W0	SLL32
PSRLI.W	SRLI32
PSRL.W.W0	SRL32
PSRAI.W	SRAI32
PSRA.W.W0	SRA32
PSSLAI.W	KSLLI32
PSRARI.W	SRAI32.u
PSSHA.W.W0	—
PSSHAR.W.W0	—
PMIN.W	SMIN32
PMINU.W	UMIN32
PMAX.W	SMAX32
PMAXU.W	UMAX32
PMSEQ.W	—
PMSLT.W	—
PMSLTU.W	—

RV64	
New instruction	Earlier equivalent
SATI	—
USATI	—
SRARI	SRAI.u
SHA	—
SHAR	—
MIN	MIN
MINU	MINU
MAX	MAX
MAXU	MAXU

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
SLX	—
SRX	—
MVM	≈ CMIX (*1)
MVMN	≈ CMIX (*2)
MERGE	≈ CMIX (*3)

RV64	
New instruction	Earlier equivalent
ABSW	≈ KABSW (*4)
CLZW	—
CLSW	—

(*1) New MVM is the same as earlier CMIX with source operand rs3 = rd.

(*2) New MVMN d,a,b = earlier CMIX d,b,d,a (that is, with CMIX's rs1, rs2, and rs3 operands = MVMN's rd, rs2, and rs1, respectively).

(*3) New MERGE d,a,b = earlier CMIX d,d,b,a (that is, with CMIX's rs1, rs2, and rs3 operands = MERGE's rs2, rd, and rs1, respectively).

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

RV32, register-pair destination			
New instruction			Earlier equivalent
PWADD.B	PWADD.H	WADD	
PWADDA.B	PWADDA.H	WADDA	*none*
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		*none*

RV32, register-pair operands			
New instruction			Earlier equivalent
PLI.DB	PLI.DH		
	PLUI.DH		*none*
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	*none*
PREDSUMU.DB	PREDSUMU.DH	

RV32, register-pair operands			
New instruction			Earlier equivalent
	PSEXTB.DH	PSEXTB.DW	
		PSEXTH.DW	*none*
	PSATI.DH	PSATI.DW	
	PUSATI.DH	PUSATI.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			
New instruction			Earlier equivalent
PADD.DB.B0	PADD.DH.H0	PADD.DW.W	
PSLL.DB.B0	PSLL.DH.H0	PSLL.DW.W	*none*
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		
New instruction		Earlier equivalent
PPACK.DH	PPACK.DW	
PPACKBT.DH	PPACKBT.DW	*none*
PPACKTB.DH	PPACKTB.DW	
PPACKT.DH	PPACKT.DW	

RV32, register-pair first source (only)			
New instruction			Earlier equivalent
PNSRLI.B	PNSRLI.H	NSRLI	
PNSRL.B.B0	PNSRL.H.H0	NSRL	*none*
PNSRAI.B	PNSRAI.H	NSRAI	
PNSRA.B.B0	PNSRA.H.H0	NSRA	
PNSRARI.B	PNSRARI.H	NSRARI	
PNSRAR.B.B0	PNSRAR.H.H0	NSRAR	
PNCLIPI.B	PNCLIPI.H	NCLIPI	
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*	—

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

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

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*	≈ KMA*pp*	(*2, 4)
PMACCSU.W.H*pp*	—	
PMACCU.W.H*pp*	—	

RV64		
New instruction	Earlier equivalent	
MUL.W*nn*	SM*pp*32	(*1)
MULSU.W*nn*	—	
MULU.W*nn*	≈ MULR64	(*5)
MACC.W*nn*	≈ KMA*pp*32	(*1, 4)
MACCSU.W*nn*	—	
MACCU.W*nn*	—	
PM2ADD.W	≈ KMDA32	(*4)
PM2ADDA.W	SMAR64	
PM2ADDSU.W	—	
PM2ADDASU.W	—	
PM2ADDU.W	—	
PM2ADDAU.W	UMAR64	
PM2ADD.WX	≈ KMXDA32	(*4)
PM2ADDA.WX	≈ KMAXDA32	(*4)
PM2SUB.W	SMDRS32	
PM2SUBA.W	≈ KMADRS32	(*4)
PM2SUB.WX	SMXDS32	(*3)
PM2SUBA.WX	≈ 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	
New instruction	Earlier equivalent
PM4ADD.B	—
PM4ADDA.B	SMAQA
PM4ADDSU.B	—
PM4ADDASU.B	SMAQA.SU
PM4ADDU.B	—
PM4ADDAU.B	UMAQA

RV64	
New instruction	Earlier equivalent
PM4ADD.H	—
PM4ADDA.H	SMALDA
PM4ADDSU.H	—
PM4ADDASU.H	—
PM4ADDU.H	—
PM4ADDAU.H	—

RV32/RV64	
New instruction	Earlier equivalent
PMULH.H.Bp	—
PMULHSU.H.Bp	—
PMHACC.H.Bp	—
PMHACCSU.H.Bp	—

RV32	
New instruction	Earlier equivalent
MULH.Hn	SMMWp (*1)
MULHSU.Hn	—
MHACC.Hn	≈ KMMAWp (*1, 3)
MHACCSU.Hn	—

RV64	
New instruction	Earlier equivalent
PMULH.W.Hp	SMMWp (*2)
PMULHSU.W.Hp	—
PMHACC.W.Hp	≈ KMMAWp (*2, 3)
PMHACCSU.W.Hp	—

(*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)

(*1) Swap the source operands.

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