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

John Hauser

March 20, 2026

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.

This version (020) differs from the previous one (018) by removing from the tables instructions that are in extensions Zba and Zbb (such as CLZ, MIN, REV8, and SH1ADD), and by adding these new instructions:

- SHL instructions (shift left or right, logical) that are unsigned versions of the SHA instructions (shift left or right, arithmetic); and
- for RV64 only, 'P'-suffix versions of PNCLIP instructions that are similar to RV32 PNCLIP instructions without shifting or rounding.

The complete set of instructions added for RV32 are:

PSSHL.HS	SSHL	PSSHL.DHS	PSSHL.DWS
PSSHLR.HS	SSHLR	PSSHLR.DHS	PSSHLR.DWS

And for RV64:

PSSHL.HS	PSSHL.WS	SHL	PNCLIPP.B	PNCLIPP.H	PNCLIPP.W
PSSHLR.HS	PSSHLR.WS	SHLR	PNCLIPUP.B	PNCLIPUP.H	PNCLIPUP.W

Also, approximate correspondences between new and earlier shift instructions are now indicated where they were not before.

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.BS	—
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
PABD.B	—
PABDU.B	—
PSABS.B	KABS8
PREDSUM.BS	—
PREDSUMU.BS	—
PABDSUMU.B	PBSAD
PABDSUMAU.B	PBSADA

RV32/RV64 New instruction	Earlier equivalent
PLI.H	—
PLUI.H	—
PADD.HS	—
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
PABD.H	—
PABDU.H	—
PSABS.H	KABS16
PREDSUM.HS	—
PREDSUMU.HS	—

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

RV64	
New instruction	Earlier equivalent
PLI.W	—
PLUI.W	—
PADD.WS	—
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.WS	—
PREDSUMU.WS	—

RV32/RV64 New instruction	Earlier equivalent
PSLLI.B	SLLI8
PSLL.BS	≈ SLL8 (*1)
PSRLI.B	SRLI8
PSRL.BS	≈ SRL8 (*1)
PSRAI.B	SRAI8
PSRA.BS	≈ SRA8 (*1)
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
PSEXT.H.B	SUNPKD820
PSATI.H	SCLIP16
PUSATI.H	UCLIP16
PSLLI.H	SLLI16
PSLL.HS	≈ SLL16 (*1)
PSRLI.H	SRLI16
PSRL.HS	≈ SRL16 (*1)
PSRAI.H	SRAI16
PSRA.HS	≈ SRA16 (*1)
PSSHL.HS	—
PSSHLR.HS	—
PSSLAI.H	KSLLI16
PSRARI.H	SRAI16.u
PSSHA.HS	≈ KSLRA16 (*2)
PSSHAR.HS	≈ KSLRA16.u (*2)
PMIN.H	SMIN16
PMINU.H	UMIN16
PMAX.H	SMAX16
PMAXU.H	UMAX16
PMSEQ.H	CMPEQ16
PMSLT.H	SCMPLT16
PMSLTU.H	UCMPLT16

(*1) The new instruction takes the shift distance from the bottom 5 bits of the second source operand, while the earlier instruction looks at only the bottom 3 bits (SLL8, SRL8, SRA8) or bottom 4 bits (SLL16, SRL16, SRA16).

(*1) The new instruction takes the signed shift distance from the bottom 8 bits (one byte) of the second source operand, while the earlier instruction looks at only the bottom 5 bits.

RV32		RV64		RV64	
New instruction	Earlier equivalent	New instruction	Earlier equivalent	New instruction	Earlier equivalent
		PSEXT.W.B	—		
		PSEXT.W.H	—		
SATI	SCLIP32	PSATI.W	SCLIP32	SATI	—
USATI	UCLIP32	PUSATI.W	UCLIP32	USATI	—
		PSLLI.W	SLLI32		
		PSLL.WS	SLL32		
		PSRLI.W	SRLI32		
		PSRL.WS	SRL32		
		PSRAI.W	SRAI32		
		PSRA.WS	SRA32		
SSHL	—	PSSHL.WS	—	SHL	—
SSHLR	—	PSSHLR.WS	—	SHLR	—
SSLAI	KSLLIW	PSSLAI.W	KSLLI32		
SRARI	SRAI.u	PSRARI.W	SRAI32.u	SRARI	SRAI.u
SSHA	≈ KSLRAW (*1)	PSSHA.WS	≈ KSLRA32 (*1)	SHA	—
SSHAR	≈ KSLRAW.u (*1)	PSSHAR.WS	≈ KSLRA32.u (*1)	SHAR	—
		PMIN.W	SMIN32		
		PMINU.W	UMIN32		
		PMAX.W	SMAX32		
		PMAXU.W	UMAX32		
MSEQ	—	PMSEQ.W	—		
MSLT	—	PMSLT.W	—		
MSLTU	—	PMSLTU.W	—		

(*1) The new instruction takes the signed shift distance from the bottom 8 bits (one byte) of the second source operand, while the earlier instruction looks at only the bottom 6 bits.

RV32/RV64	
New instruction	Earlier equivalent
PPAIRE.B	—
PPAIREO.B	—
PPAIROE.B	—
PPAIRO.B	—

RV32/RV64	
New instruction	Earlier equivalent
PPAIRE.H	PKBB16 (*1)
PPAIREO.H	PKTB16 (*1)
PPAIROE.H	PKBT16 (*1)
PPAIRO.H	PKTT16 (*1)
PACK	PACK

RV64	
New instruction	Earlier equivalent
PPAIREO.W	PKTB32 (*1)
PPAIROE.W	PKBT32 (*1)
PPAIRO.W	PACKU

(*1) Swap the source operands.

RV64	
New instruction	Earlier equivalent
ZIP8P	—
ZIP8HP	—
UNZIP8P	—
UNZIP8HP	—
PNCLIPP.B	—
PNCLIPUP.B	—

RV64	
New instruction	Earlier equivalent
REV16	—
ZIP16P	—
ZIP16HP	—
UNZIP16P	—
UNZIP16HP	—
PNCLIPP.H	—
PNCLIPUP.H	—

RV64	
New instruction	Earlier equivalent
PNCLIPP.H	—
PNCLIPUP.H	—

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

RV64	
New instruction	Earlier equivalent
ABSW	≈ KABSW (*4)
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.BS	PWSLL.HS	WSLL	
PWSLAI.B	PWSLAI.H	WSLAI	
PWSLA.BS	PWSLA.HS	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		
PABD.DB	PABD.DH		
PABDU.DB	PABDU.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.DBS	PREDSUM.DHS	*none*
PREDSUMU.DBS	PREDSUMU.DHS	

RV32, register-pair operands			
New instruction			Earlier equivalent
	PSEXT.DH.B	PSEXT.DW.B	
		PSEXT.DW.H	*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.DBS	PADD.DHS	PADD.DWS	
PSLL.DBS	PSLL.DHS	PSLL.DWS	*none*
PSRL.DBS	PSRL.DHS	PSRL.DWS	
PSRA.DBS	PSRA.DHS	PSRA.DWS	
	PSSHL.DHS	PSSHA.DWS	
	PSSHLR.DHS	PSSHLR.DWS	
	PSSHA.DHS	PSSHA.DWS	
	PSSHAR.DHS	PSSHAR.DWS	

RV32, register-pair operands		
New instruction		Earlier equivalent
PPAIRE.DB	PPAIRE.DH	
PPAIREO.DB	PPAIREO.DH	*none*
PPAIROE.DB	PPAIROE.DH	
PPAIRO.DB	PPAIRO.DH	

RV32, register-pair first source (only)			
New instruction			Earlier equivalent
PNSRLI.B	PNSRLI.H	NSRLI	
PNSRL.BS	PNSRL.HS	NSRL	*none*
PNSRAI.B	PNSRAI.H	NSRAI	
PNSRA.BS	PNSRA.HS	NSRA	
PNSRARI.B	PNSRARI.H	NSRARI	
PNSRAR.BS	PNSRAR.HS	NSRAR	
PNCLIPI.B	PNCLIPI.H	NCLIPI	
PNCLIP.BS	PNCLIP.HS	NCLIP	
PNCLIPRI.B	PNCLIPRI.H	NCLIPRI	
PNCLIPR.BS	PNCLIPR.HS	NCLIPR	
PNCLIPIU.B	PNCLIPIU.H	NCLIPIU	
PNCLIPU.BS	PNCLIPU.HS	NCLIPU	
PNCLIPRIU.B	PNCLIPRIU.H	NCLIPRIU	
PNCLIPRU.BS	PNCLIPRU.HS	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*nn*	—
PMQRACC.W.H*nn*	—

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*nn*	—
PMULSU.H.B*nn*	—
PMULU.H.B*nn*	—

RV32		
New instruction	Earlier equivalent	
MUL.H*nn*	SM*pp*16	(*1)
MULSU.H*nn*	—	
MULU.H*nn*	—	
MACC.H*nn*	≈ KMA*pp*	(*1, 3)
MACCSU.H*nn*	—	
MACCU.H*nn*	—	

RV32/RV64		
New instruction	Earlier equivalent	
PM2ADD.H	≈ KMDA	(*3)
PM2ADDA.H	≈ KMADA	(*3)
PM2ADDSU.H	—	
PM2ADDASU.H	—	
PM2ADDU.H	—	
PM2ADDAU.H	—	
PM2ADD.HX	≈ KMXDA	(*3)
PM2ADDA.HX	≈ KMAXDA	(*3)
PM2SADD.H	KMDA	
PM2SADD.HX	KMXDA	
PM2SUB.H	SMDRS	
PM2SUBA.H	≈ KMADRS	(*3)
PM2SUB.HX	SMXDS	(*2)
PM2SUBA.HX	≈ KMAXDS	(*2, 3)

RV64		
New instruction	Earlier equivalent	
PMUL.W.H*nn*	SM*pp*16	(*1)
PMULSU.W.H*nn*	—	
PMULU.W.H*nn*	—	
PMACC.W.H*nn*	≈ KMA*pp*	(*1, 3)
PMACCSU.W.H*nn*	—	
PMACCU.W.H*nn*	—	

RV64		
New instruction	Earlier equivalent	
MUL.W*nn*	SM*pp*32	(*1)
MULSU.W*nn*	—	
MULU.W*nn*	≈ MULR64	(*4)
MACC.W*nn*	≈ KMA*pp*32	(*1, 3)
MACCSU.W*nn*	—	
MACCU.W*nn*	—	
PM2ADD.W	≈ KMDA32	(*3)
PM2ADDA.W	SMAR64	
PM2ADDSU.W	—	
PM2ADDASU.W	—	
PM2ADDU.W	—	
PM2ADDAU.W	UMAR64	
PM2ADD.WX	≈ KMXDA32	(*3)
PM2ADDA.WX	≈ KMAXDA32	(*3)
PM2SUB.W	SMDRS32	
PM2SUBA.W	≈ KMADRS32	(*3)
PM2SUB.WX	SMXDS32	(*2)
PM2SUBA.WX	≈ KMAXDS32	(*2, 3)

(*1) For the sub-elements to select from each source element, the earlier instruction has p being 'B' or 'T', corresponding to an n of '0' or '1'.

(*2) Swap the source operands.

(*3) The new instruction does not saturate the addition(s), while the earlier instruction does.

(*4) 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.Bn	—
PMULHSU.H.Bn	—
PMHACC.H.Bn	—
PMHACCSU.H.Bn	—

RV32	
New instruction	Earlier equivalent
MULH.Hn	SMMWp (*1)
MULHSU.Hn	—
MHACC.Hn	≈ KMMAWp (*1, 2)
MHACCSU.Hn	—

RV64	
New instruction	Earlier equivalent
PMULH.W.Hn	SMMWp (*1)
PMULHSU.W.Hn	—
PMHACC.W.Hn	≈ KMMAWp (*1, 2)
PMHACCSU.W.Hn	—

(*1) For the sub-elements to select from the second operand, the earlier instruction has p being 'B' or 'T', corresponding to an n of '0' or '1'.

(*2) 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