CAHPv3																					EX Opcode			Instruction Category		24bit Inst FLAG					
Notes	Ops	How it works	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	inA	inB	op	inst	Test
M-Instruction																															
	lw rd, simm10(rs)	rd <- [rs + simm10]	simm10[7:0]								rs				rd				simm10[9:8]		0	1	0	1	0	1	rs	simm10	ADD	0	o
	lb rd, simm10(rs)	rd <- [rs + simm10]	simm10[7:0]								rs				rd				simm10[9:8]		1	0	0	1	0	1	rs	simm10	ADD	1	o
	lbu rd, simm10(rs)	rd <- [rs + simm10]	simm10[7:0]								rs				rd				simm10[9:8]		0	0	0	1	0	1	rs	simm10	ADD	1	o
	sw rs, simm10(rd)	[rd + simm10] <- rs	simm10[7:0]								rd				rs				simm10[9:8]		x=0	1	1	1	0	1	rd	simm10	ADD	0	o
	sb rs, simm10(rd)	[rd + simm10] <- rs	simm10[7:0]								rd				rs				simm10[9:8]		0	0	1	1	0	1	rd	simm10	ADD	1	o
	li rd, simm10	rd <- simm10	simm10[7:0]								x=0	x=0	x=0	x=0	rd				simm10[9:8]		1	1	0	1	0	1		simm10	MOV	1	o
R-Instruction																															
	add rd, rs1, rs2	rd <- rs1 + rs2	x=0	x=0	x=0	x=0	rs2				rs1				rd				x=0	0	0	0	0	0	0	1					o
	sub rd, rs1, rs2	rd <- rs1 - rs2	x=0	x=0	x=0	x=0	rs2				rs1				rd				x=0	0	0	0	1	0	0	1					o
	and rd, rs1, rs2	rd <- rs1 & rs2	x=0	x=0	x=0	x=0	rs2				rs1				rd				x=0	0	0	1	0	0	0	1					o
	xor rd, rs1, rs2	rd <- rs1 ^ rs2	x=0	x=0	x=0	x=0	rs2				rs1				rd				x=0	0	0	1	1	0	0	1					o
	or rd, rs1, rs2	rd <- rs1 | rs2	x=0	x=0	x=0	x=0	rs2				rs1				rd				x=0	0	1	0	0	0	0	1					o
	lsl rd, rs1, rs2	rd <- rs1 << rs2	x=0	x=0	x=0	x=0	rs2				rs1				rd				x=0	0	1	0	1	0	0	1					o
	lsr rd, rs1, rs2	rd <- rs1 >> rs2	x=0	x=0	x=0	x=0	rs2				rs1				rd				x=0	0	1	1	0	0	0	1					o
	asr rd, rs1, rs2	rd <- rs1 >>> rs2	x=0	x=0	x=0	x=0	rs2				rs1				rd				x=0	0	1	1	1	0	0	1					o
I-Instruction																															
	addi rd, rs1, simm10	rd <- rs1 + simm10	simm10[7:0]								rs1				rd				simm10[9:8]		0	0	0	0	1	1	rs1	simm10		2	o
	andi rd, rs1, simm10	rd <- rs1 & simm10	simm10[7:0]								rs1				rd				simm10[9:8]		0	1	0	0	1	1	rs1	simm10		3	o
	xori rd, rs1, simm10	rd <- rs1 ^ simm10	simm10[7:0]								rs1				rd				simm10[9:8]		0	1	1	0	1	1	rs1	simm10		3	o
	ori rd, rs1, simm10	rd <- rs1 | simm10	simm10[7:0]								rs1				rd				simm10[9:8]		1	0	0	0	1	1	rs1	simm10		3	o
	lsli rd, rs1, uimm4	rd <- rs1 << uimm4	x=0	x=0	x=0	x=0	uimm4[3:0]				rs1				rd				x=0	x=0	1	0	1	0	1	1	rs1	uimm4		3	o
	lsri rd, rs1, uimm4	rd <- rs1 >> uimm4	x=0	x=0	x=0	x=0	uimm4[3:0]				rs1				rd				x=0	x=0	1	1	0	0	1	1	rs1	uimm4		3	o
	asri rd, rs1, uimm4	rd <- rs1 >>> uimm4	x=0	x=0	x=0	x=0	uimm4[3:0]				rs1				rd				x=0	x=0	1	1	1	0	1	1	rs1	uimm4		3	o
J-Instruction																															
	beq rs1, rs2, simm10	if rs1 == rs2 then PC <- PC + simm10	simm10[7:0]								rs1				rs2				simm10[9:8]		0	0	1	1	1	1					o
	bne rs1, rs2, simm10	if rs1 != rs2 then PC <- PC + simm10	simm10[7:0]								rs1				rs2				simm10[9:8]		1	0	1	1	1	1					o
	blt rs1, rs2, simm10	if rs1 < rs2 then PC <- PC + simm10	simm10[7:0]								rs1				rs2				simm10[9:8]		1	1	0	1	1	1					o
	bltu rs1, rs2, simm10	if rs1 < rs2 then PC <- PC + simm10	simm10[7:0]								rs1				rs2				simm10[9:8]		0	1	0	1	1	1					o
	ble rs1, rs2, simm10	if rs1 <= rs2 then PC <- PC + simm10	simm10[7:0]								rs1				rs2				simm10[9:8]		1	1	1	1	1	1					o
	bleu rs1, rs2, simm10	if rs1 <= rs2 then PC <- PC + simm10	simm10[7:0]								rs1				rs2				simm10[9:8]		0	1	1	1	1	1					o
実装しない	j simm16	PC <- PC + simm16	simm16[15:0]																x=0	x=0	x=0	0	0	1	1	1					
実装しない	jal simm16	RA <- PC + 4, PC <- PC + simm16	simm16[15:0]																x=0	x=0	x=0	0	0	1	1	1					
Notes	Ops	How it works	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	inA	inB	op		
	16bit Length Instruction																														
M-Instruction																															
	lwsp rd, uimm7(sp)	rd <- [sp + uimm7]									uimm7[4:1]				rd				uimm7[6:5]		0	1	0	1	0	0	sp	uimm7	ADD	4	o
	swsp rs, uimm7(sp)	[sp + uimm7] <- rs									uimm7[4:1]				rs				uimm7[6:5]		0	1	1	1	0	0	sp	uimm7	ADD	4	o
	lsi rd, simm6	rd <- simm6									simm6[3:0]				rd				simm6[5:4]		1	1	0	1	0	0		imm	MOV	5	o
	lui rd, simm6	rd <- (simm6 << 10)									simm6[3:0]				rd				simm6[5:4]		0	0	0	1	0	0		imm	MOV	6	o
R-Instruction																															
	mov rd, rs	rd <- rs									rs(rs2)				rd(rs1)				1	1	0	0	0	0	0	0					o
	add2 rd, rs	rd <- rd + rs									rs(rs2)				rd(rs1)				1	0	0	0	0	0	0	0					o
	sub2 rd, rs	rd <- rd - rs									rs(rs2)				rd(rs1)				1	0	0	0	1	0	0	0					o
	and2 rd, rs	rd <- rd & rs									rs(rs2)				rd(rs1)				1	0	0	1	0	0	0	0					o
	xor2 rd, rs	rd <- rd ^ rs									rs(rs2)				rd(rs1)				1	0	0	1	1	0	0	0					o
	or2 rd, rs	rd <- rd | rs									rs(rs2)				rd(rs1)				1	0	1	0	0	0	0	0					o
	lsl2 rd, rs	rd <- rd << rs									rs(rs2)				rd(rs1)				1	0	1	0	1	0	0	0					o
	lsr2 rd, rs	rd <- rd >> rs									rs(rs2)				rd(rs1)				1	0	1	1	0	0	0	0					o
	asr2 rd, rs	rd <- rd >>> rs									rs(rs2)				rd(rs1)				1	0	1	1	1	0	0	0					o
I-Instruction																															

	lsli2 rd, uimm4	rd <- rd << uimm4									uimm4[3:0]				rd(rs1)				x=0	x=0	1	0	1	0	1	0				6	o
	lsri2 rd, uimm4	rd <- rd >> uimm4									uimm4[3:0]				rd(rs1)				x=0	x=0	1	1	0	0	1	0				6	o
	asri2 rd, uimm4	rd <- rd >>> uimm4									uimm4[3:0]				rd(rs1)				x=0	x=0	1	1	1	0	1	0				6	o
	addi2 rd, simm6	rd <- rd + simm6									simm6[3:0]				rd(rs1)				simm6[5:4]		0	0	0	0	1	0				5	o
	andi2 rd, simm6	rd <- rd & simm6									simm6[3:0]				rd(rs1)				simm6[5:4]		0	1	0	0	1	0				6	o
	J-Instruction																														
	jalr rs	RA <- PC + 2, PC <- rs									x=0	x=0	x=0	x=0	rs(rs1)				x=0	x=0	x=0	1	0	1	1	0	pc	2	ADD	7	o
	jr rs	PC <- rs									x=0	x=0	x=0	x=0	rs(rs1)				x=0	x=0	x=0	0	0	1	1	0					o
	js simm11	PC <- PC + simm11									simm11[10:0]											0	1	1	1	0					o
	jsal simm11	RA <- PC + 2, PC <- PC + simm11									simm11[10:0]											1	1	1	1	0	pc	4	ADD	8	o
	Others																														
	nop										x=0	x=0	x=0	x=0	x=0	x=0	x=0	x=0	0	x=0	x=0	x=0	x=0	0	0	0					o