bits	name		Big Endian Network Order	
3	primary op		An carries	
2	dest Addr mode		Dn modulo size, high no effect	
3	dest reg			
1	secondary op	quicks and extra	Double(1)/Float(0) Default is D	
2	size	L is default size		
2	src Addr mode	5	0 equals 32	
3	src reg	quick	branch signed from PC+1	
Addr mode			size	
#00	An	Not CFR src	#00	.W
#01	(An)		#01	.L
#10	Dn	Not CFR src	#10	.Q
#11	Special		#11	.D/.F
Special				
#000	PC	Not float	Not CFR src	
#001	(PC+)/(-SP)	special write	For PCX subroutine	
#010	IP	Not float	Not CFR src	
#011	(IP+)/(-RP)	special write	For IPX threaded code	
#100	SP	Not float	Not CFR src	
#101	(SP+)/(-SP)			
#110	RP	Not float	Not CFR src	
#111	(RP+)/(-RP)			
ops	D		F	
#000	MOV	No dest read	MVQ	quick src args
#001	ADD		ADQ	unsigned 5 bit
#010	MLL	Multiply lower	MLQ	for compact
#011	SUB		SBQ	code
#100	CFR (specials)	No dest read	BPL	quick branch
#101	AND		BMI	0 and -1 do sign
#110	ORR		BZE	of 16 bit
#111	XOR		BNZ	displacement
D/F ops (size)	D/F			
#000	FDV	Divide		
#001	FAD			
#010	FML			
#011	FSB			
#100	FBP	quick branch		
#101	FBM	0 and 1 do sign		
#110	FBD	of 16 bit	Branch denormal zero	

#111	FBI	displacement	Branch infinity	
For efficient floats with own register set use vectors				

CFR src					
An	halves	Reg used as tertiary select operation			
Dn	fulls				
PC	PCX	fast exchange			
IP	IPX	of registers			
SP	SPX	improves code			
RP	RPX	good stuff			
The src MUST be an address for CFR to be CFR					
halves An					
Blocks of 8	B,W,L = half				
#000	EXH	Exchange halves			
#001	SXL	Sign extend low			
#010	INU	Increase upper			
#011	DCU	Decrease upper			
#100	SLT	Shift left Q half			
#101	SRT	Shift right Q half			
#110	ADU	Add upper			
#111	SBU	Subtract upper			
fulls Dn					
Blocks of 8	W,L,Q				
#000	ABS	Absolute			
#001	SGN	Sign to 0 or -1			
#010	NOT	Not			
#011	CPL	Complement			
#100	WTO (group)	Write target override			
#101	LSB	least significant bit (modulo 2)			
#110	ASR	Arith shift right			
#111	LSR	Logic shift right			
	W	L	Q		
WTO group	UCC	UCE	WTO		
WTO is set write target override for following instruction for 3 operands					
Only affects following instruction if at all					
When used with MOV, MVQ, CFR but not specials except WTO makes a reserved nop pair					
These reserved nop pairs may be assigned other function					
UCC is Unicode compress. W to a L with or joining for surrogate half					
UCE is Unicode expand. L to a W with bit 27 set to make hi surrogate					
WTO a MOV c,b becomes a = b/c for example				DIV	

Similar with WTO MVQ all done unsigned				DVQ	
WTO branch becomes conditional register load with address (overrides PC write)					
WTO then					
CFR	GTA (P/H)	Get translated address			
WTO	CSM	Checksum check CSM keyexpect,delta			
CSM is an execution checksum calculated and acted upon					
The keyexpect can be put south of the checked code nesting include of check ok routine					
As can be CPU ID specific can make code for CPU dependent on CPU					
Check sum garuntee of execution and delta of last as initial post CSM					
A checksum watchdog might be useful					

D (dst\src)	PC vectors	IP	SP	RP	
PC	BRK	Reserved	Reserved extension opcodes		
IP	TRP	system design	8 * 8 = 64		
SP	HPR (P)	supervisor (P)			
RP	CIV (H)	hypervisor (H)			
F	PC	IP	SP	RP	
PC	COM (0 to 31)	User opcode	Reserved vector prefix opcodes		
IP	compressed	extensions	8 * 8 = 64		
SP	pair	eval as nop			
RP	extensions	if not used (32)			
register D	PC	IP	SP	RP	
is src		FCN (float constant) not branches			
is dest	@BRK	@TRP (P)	@HVR (H)	@CIV (I)	Set d64
register F	PC	IP	SP	RP	
is src		FCN (float constant) not branches			
is dest		@COM			Set d32
Blocks of 8	supervisor (P)		hypervisor (H)		
#000	SNP	super nop	HNP	hyper nop	
#001	EVM	enable virt mem	EIN (I)	enable interrupt	
#010	DVM	disable virt mem	DIN (I)	disable interrupt	
#011	WTI	await interrupt	CLI (I)	clear interrupt	
#100					
#101	CLS	clear supervisor	CLH	clear hypervisor	
#110	RSS	reset super	RSH	reset hyper	
#111	SIL	illegal super	HIL	illegal hyper	
HPR is hypervisor trap					
BRK is user mode breakpoint trap					
CIV is set call interrupt vector (interrupt 0 is page fault)					
COM instruction pairs last can be tail recursive					
COM first can not be					
This makes (@COM n COM m d16) a spare opcode space					
@ opcodes can't use PC, IP, SP, RP as src					
WTO makes all these nop pairs					
Only 32 float/double constants like pi					

(@COM n (COM m d16)) spare opcode space				
#nnn11nn0 011ssss opcode form points to				
#mmm11mm0 011000				
#dddddddd dddddddd				
Difficult to use directly				
Only first of pair, branches delay second as delay slot.				
Second after branch may or may not speculative execute and commit				
Allow setting up and special COM?				
#mmm11mm0 011000				
#dddddddd dddddddd		is the instruction replaced by profile COM		
mmm	profile counter number to increase			
mm	recall to register except PC			
If register is PC just increment profile counter				
Any immediates are ordered in the main instruction stream				
(BRK/TRP/HPR/CIV m d16) spare opcode space				
#mmm11mm0 111000				
#dddddddd dddddddd				
Except when mm is PC nnn is register special below				
#mmm				
#000	Processor serial number			
#001	Clock counter for cycles since power on (timing)			
#010	Clocks stalled since on for efficiency measure			
#011	Clock with boost cycles since on			
#100	Branch miss count since on			
#101	Branch hit count since on			
#110	Data cache miss since on			
#111	Instruction cache miss since on			
When nn is PC clear profile register nnn				
The register specials are only cleared via power cycle				

NOP stability code watermark set					
src,dst	PC,PC	PC,other	other,PC	PC	
MLL	WRP		ONG		
MLQ			RDP	RDP as left	
halves			CLR	CLR as left	An
fulls			CLR	CLR as left	Dn
BPL				32	
BMI				32	
BZE				32	
BNZ				32	
AND	1	GUP (P)	SUP (P)		
ORR	1	GSP (H)	SSP (H)		
XOR	1	GHP (I)	SHP (I)		
131	3	0	0	128	
All evaluate as nop unless assigned					
Possible extension opcodes * sizes			393	16 bit	
WTO makes all these nop pairs					
Add in WTO any nop pair versions			12969	16 bit + 32 bit	
And WTO any for D/F with PC, IP, SP, RP					
D16	16	8	128		
F16	16	8	128		
DS4	2	224	448		
DD4	4	224	896		
FS4	2	224	448		
FD4	4	224	896		
			97152		
Reg - 4	28				
OPs per block	8		110121	Opcode prefixes	
Excludes some WTO PC int and WTO PC,IP,SP,RP float-double				Still nops	
All these express as reserved double nops					
All on this page are reserved					
WRP is write prefix WRP d16 as nop unless used for d16 needing xtra write location					
RDP d16 as nop unless used for d16 needing extra read					
WTO RDP d16 is valid if d16 has alternate write target					
RDP WTO is nop as WTO will be d16 yet to be defined					
ONG is on goto for dense modulo switch jumps following instruction, zero based					
CLR is clear register using read decoder with gated reset					
Can execute fast and drop from pipeline as a MVQ 0 internally					
SHP is set hypervisor paging and may be used from interrupt only					

SUP and SSP set user an supervisor paging			
The Gxx versions get the page table pointer			
Size on page pointer do short forms of lower word(s)			

There are many other nops. Mainly on src,dest choices					
DVQ #1,anydividand,anyquotient? MLQ #1,any?					
Last mnemonic operand of 3 is WTO operand					
Excluding the above 2 all parameterized nops are not reserved					
They remain as power crypto differential nop tools and could assist automated coding					
DVQ #1	SRT	Square root			
MLQ #1	STB	Set bit number rest zero modulo size			
This completes the integer arithmetic operations					
The reads can be parallel, see pipeline below					
Pipeline					
Fetch/Decode	Read Src+	Read Dst+	Execute-	Write Dst Commit	
	WTO				nop
INS					nop
		WTO			nop
	INS				R2S+ (nop?)
			WTO		X1-
		INS			R2D+
				WTO	W1 (pass back)
			INS		X2- (pos. void)
					nop
				INS	W2 (commit)
INS is R2S+ R2D+ (-W1 or -W2)					
PC,IP,SP and RP speculate value for commit					
Write overrides + or -					
Red box potential sum zero pre post					