EPIPHANY ARCHITECTURE QUICK REFERENCE CARD

INSTRUCTION SET

ASSEMBLER	ACTION
BRANCHES	
B<COND> <simm24>	If <COND> PC=PC+<simm24> Else PC = next instr
BL <simm24>	PC=PC + <simm24>
JR RN	PC=RN
JALRN RN	PC=RN, LR= next instruction
LOAD/STORE	
LDR{size} RD,[RN, #+/-<offset>]	RD=[RN+/-offset, size]
STR{size} RD,[RN, #+/-<offset>]	[RN+/-offset, size]=RD
LDR{size} RD,[RN], #+/-<offset>	RD=[RN];RN=RN+/-offset
STR{size} RD,[RN],#+/-<offset>	[RN]=RD;RN=RN+/-offset
TESTSET RD, [RN, RM]	if ([RN+RM])RD= [RN+RM]; Else [RN+RM]=RD;RD=0;
INTEGER	
ADD RD,RN,<OP2>	RD=RN + OP2
SUB RD,RN,<OP2>	RD=RN – OP2
ASR RD,RN, <OP2>	RD=RN >>> OP2
LSR RD, RN, <OP2>	RD=RN >> OP2
LSL RD, RN, <OP2>	RD=RN << OP2
ORR RD, RN, RM	RD=RN | RM
AND RD, RN, RM	RD=RN & RM
EOR RD, RN, RM	RD= RN ^ RM
BITR RD, RN	RD = BIT-REVERSE (RN)
FLOATING POINT / SECONDARY INTEGER	
FADD/IADD RD, RN, RM	RD = RN + RM
FSUB/ISUB RD, RN, RM	RD = RN – RM
FMUL/IMUL RD, RN, RM	RD = RN * RM
FMADD/IMADD RD, RN, RM	RD += RN * RM
FMSUB/IMSUB RD, RN, RM	RD -= RN * RM
FABS RD, RN	RD=ABS (RN)
FIX RD, RN	RD=FIX (RN)
FLOAT RD,RN	RD=FLOAT (RN)
MOVE	
MOV RD, <imm16>	RD = <imm16>
MOVT RD, <imm16>	RD = RD | (<imm16> <<16)
MOV<COND> RD, RN	if(<COND>) RD=RN
MOVTS RD, RN	RD=RN
MOVFS RD, RN	RD=RN
CONTROL	
NOP	None
IDLE	PC=PC until IRQ
RTS	PC=LR
RTI	PC=IRET
GID/GIE	Disable/Enable all interrupts
BKPT/MBKPT	Breakpoint/(Multiore bkpt)
SYNC	Broadcast ILAT[0]=1
WAND	Status[2]=1;IDLE
TRAP	Halts program

NOTES:

<COND>={EQ,NE,GTU,GTEU,LTEU,LTU, GT,GTE,LT,LTE}	
RD,RN,RM = Registers	<size>=Data size (B,H,L,D)
<OP2>=RM or <simm11>	<imm5>=0 to 31
<offset>=RM | <imm11>	<imm11>=0 to 2047.
<simm24>-8,388,608 to 8,388,607	<simm11>= -1024 to +1023.

MEMORY MAP

ADDR	FUNCTION
00000000	Local core IVT
00000028	Local core reserved space
00000100	Start of user SRAM
000f0000	MMR
00100000	Core (0,1)
00200000	Core (0,2)
…	…
fff00000	Core(63,63)

INTERRUPTS

0	Sync / start / "reset"
4	Software exception
8	Memory exception
c	Timer0 expired
10	Timer1 expired
14	Multicore message
18	DMA0 completion
1c	DMA1 completion
20	Wired-and multicore signal
24	User interrupt

STATUS REGISTER

[0]	Core active
[1]	Interrupts disabled
[3]	Wand flag
[7:4]	{av,ac,an,az} flags
[11:8]	{bv,bc,bn,bz} flags
[15:12]	{bus,bvs,bis,avs} flags
[19:16]	Exception cause

CONFIG REGISTER

[0]	Rounding mode
[3:1]	FPU exception enables
[11:4]	Timer configuration
[15:12]	NOC routing mode

DMACONFIG REGISTER

[0]	DMA channel enable
[1]	Master mode
[2]	Chain mode
[3]	Startup bit
[4]	IRQ output enable
[6:5]	Transfer size
[10]	Message mode
[15:12]	Stride shifting

DMA DESCRIPTOR

B1:0	DMACONFIG
B3:2	Next pointer
B5:4	DMASTRIDE (inner src)
B7:6	DMASTRIDE (inner dst)
B9:8	DMACOUNT (inner)
B11:10	DMACOUNT (outer)
B13:12	DMASTRIDE (outer src)
B15:14	DMASTRIDE (outer dst)
B19:16	DMASRCADDR
B23:20	DMADSTADDR

REGISTERS

NAME	ADDR	FUNCTION
R3:R0	00c:000	Arg4:1 (caller)
R8:R4	020:010	Var5:1 (callee)
R9	024	SB: Stack base (callee)
R10	028	SL: Stack limit (callee)
R11	02c	FP: Frame pointer (callee)
R12	030	Scratch (caller)
R13	034	SP: Stack pointer
R14	038	LR: Link register (callee)
R15	03c	Scratch (callee)
R27:R16	06c:040	General (caller)
R31:R28	07c:070	Constants
R43:R32	0ac:080	General (callee)
R63:R44	0fc:0b0	General (caller)
CONFIG	400	Rounding mode
STATUS	404	Core active
PC	408	Program counter
DEBUGSTATUS	40c	Core debug state status
LC	414	HW loop counter
LS	418	HW loop start address
LE	41c	HW loop end address
IRET	420	Interrupt return address
IMASK	424	Interrupt masking register
ILAT	428	Interrupt latch register
ILATST	42C	Interrupt set alias
ILATCL	430	Interrupt clear alias
IPEND	434	Interrupt pending register
CTIMER0	438	Timer0
CTIMER1	43c	Timer1
FSTATUS	440	Status register write alias
DEBUGCMD	448	Debug command inputs
DMA{0,1}CONFIG	500/520	DMA channel enable
DMA{0,1}0STRIDE	504/524	DMA {destination, source} stride
DMA{0,1}COUNT	508/528	DMA {outer,inner} loop counter
DMA{0,1}SRCADDR	50c/52c	DMA current source address
DMA{0,1}DSTADDR	510/530	DMA current destination address
DMA{0,1}AUTO0	514/534	DMA streaming buffer (lower)
DMA{0,1}AUTO1	518/538	DMA streaming buffer (upper)
DMA0STATUS	51c/53c	DMA state
MEMSTATUS	604	Memory exception status
MEMPROTECT	608	Memory protection configuration
MESHCONFIG	700	Mesh node configuration
COREID	704	Coreid {row,col}
MULTICAST	708	Multicast ID
RESETCORE	70c	Soft reset register

PIPELINE

1:FE	PC sent to memory
2:IM	Instruction returned from memory
3:DE	Instructions decoded
4:RA	Operands read from register file
5:E1	Integer instructions completed
6:E2	Load data returns;FPU stage
7:E3	FPU write back for truncate mode
8:E4	FPU rounding and write back