
Chapter 3

Tables

Table 3.1 STATIC FIELDS PER OP ENTRY

Field Identifier	Bits/Entry
Type[2:0]	3
Imm	1
Src1Reg[4:0]	5
Src2Reg[4:0]	5
DestReg[4:0]	5
SrcStReg[4:0]	5
Src1BM[1:0]	2
Src2BM[1:0]	2
Src12BM[2]	1
SrcStBM[2:0]	3
OpInfo[12:0]	13

Table 3.2 DYNAMIC FIELDS PER OP ENTRY

FIELD IDENTIFIER	BITS/ENTRY
State[3:0]	4
ExecX	1
DestBM[2:0]	3
DestVal[31:0]	32
StatMod[3:0]	4
StatVal[7:0]	8
OprndMatch LUsrc1	1
OprndMatch LUsrc2	1
OprndMatch SUsrc1	1
OprndMatch SUsrc2	1
OprndMatch SUsrcSt	1
OprndMatch RUXsrc1	1
OprndMatch RUXsrc2	1
OprndMatch RUYsrc1	1
OprndMatch RUYsrc2	1
DBN[3:0]	4

Table 3.3 OPQUAD FIELDS PER OPQUAD

Field Identifier	Static/Dynamic	Bits/OpQuad
Emcode	static	1
Eret	static	1
FaultPC[31:0]	static	32
BPTInfo[14:0]	static	15
RASPtr[2:0]	static	3
LimViol	dynamic	1
OpQV	dynamic	1
OpQFpOp	static	1
Ilen0[2:0]	static	3
Smc1stAddr	static	20
Smc1stPg	static	20
SMC2ndAddr	static	20
Smc2ndPg	static	20

Table 3.4 OP TYPE SPECIFIED BY THE TYPE FIELD

Type(2:0)	Type of Op
000	a SpecOp—not issued to an execution unit
010	a LdOp—issued to the Load Unit
100	a StOp—issued to the Store Unit
101	a StOp that references memory—issued to the Store Unit
110	a RegOp that can only be executed by RUX—issued to RUX
111	a RegOp that can be executed by RUX or RUY—issued to either RUX or to RUY

Table 3.5 OPINFO DATA FOR A REGOP

Op Template	Description
Type[5:0]	copy of the original Op Type field
Ext[3:0]	copy of the original Op Ext field
R1	copy of the original Op R1 field
DataSz[1:0]	effective data size of the Op (one, two, or four bytes)

Table 3.6 OPINFO DATA FOR A LDSTOP

Op template	Description
Type[3:0]	copy of the original Op Type field
ISF[1:0]	copy of the original Op ISF field
Seg[3:0]	copy of the original Op Seg field
DataSz[1:0]	effective data size for the memory transfer
AddrSz	effective address size for the address calculation

Table 3.7 OPINFO DATA FOR A SPECOP

Op template	Description
Type[3:0]	copy of the original Op Type field
CC[4:0]	copy of the original Op CC field

Table 3.8 INTERPRETATIONS OF THE STATE FIELD

S3	S2	S1	S0	Indicates the Op is
0	0	0	0	unissued / not yet issued
0	0	0	1	in operand fetch stage
0	0	1	1	in execution stage 1
0	1	1	1	in execution stage 2
1	1	1	1	completed

Table 3.9 STATUS FLAGS GROUPS SPECIFIED BY THE STATMOD FIELD

StatMod bit	Status Flags Groups that can be Modified by RegOps
3	{EZF, ECF}
2	{OF}
1	{SF, ZF, AF, PF}
0	{CF}

Table 3.10 ADDITION OF NEW ROW TO TABLE 2.43

Figure 2.16 and Figure 2.19	Figure 2.12, Figure 2.14 and Figure 2.18
Not shown in either figure	Op Issue Stage
S0	Operand Fetch Stage
S1	Execution Stage 1
S2	Execution Stage 2
C	Commit Stage

Table 3.11 POTENTIAL CONFLICTS WHEN READING/WRITING OPERAND/RESULT VALUES

1st command	2nd command	Notation or Terminology	Example of Potential Conflict
Read	Read	RAR Read After Read	In this case, both commands use the same result, produced by some previous command. There is no conflict and the two commands can be executed in any order.
Read	Write	WAR Write After Read	In this case, the 1st command reads a result produced by some previous command from a specific storage location and then the 2nd command writes to the same storage location. A conflict exists if the 2nd command writes a new result to the storage location in question before the 1st command reads the older, previous result value from it.
Write	Read	RAW Read After Write	In this case, the 2nd command uses the result produced by the 1st command. A conflict exists if the 2nd command reads the result's storage location before the 1st command has written its result to the storage location in question as the 2nd command will be reading the wrong result value—the result produced by some other command.
Write	Write	WAW Write After Write	In this case, the 1st command writes its result value to a specific storage location and then the 2nd command writes its result to the same storage location. A conflict exists if the 2nd command's write occurs before the 1st command's write as the storage location is left with the wrong result. Subsequent reads to the storage location will read this wrong value.