Appendix C: iWarp Include Files

The following include files can be found in /iwarp/include/f77 and should be included by using include "f77/*name.h*":

f77/gates.h	definitions of GATE0 and GATE1 for communication intrinsics
f77/iwarp_cc.h	safe (asynchronous) iWarp condition codes for _COND and _CONDCALL
f77/iwarp_cc77.h	same as f77/iwarp_cc.h, but with short names that can be used when VMS extensions are not enabled
f77/iwarp_math_cc.h	unsafe (synchronous) iWarp condition codes for _COND and _CONDCALL
f77/iwarp_math_cc77.h	same as f77/iwarp_math_cc.h, but with short names that can be used when VMS extensions are not enabled
f77/regnums.h	register name definitions for asm-inlines
f77/regnums77.h	same as f77/regnums.h, but with short names that can be used when VMS extensions are not enabled

Safe/Asynchronous Condition Codes

Table C-1 a complete list of the condition codes associated with asynchronous conditions. All these codes are considered safe for use in _COND and _CONDCALL constructs. The condition code names listed here are defined by include "f77/iwarp_cc.h" or include "f77/iwarp_cc77.h".

Table C-1: Asynchronous Condition Codes

Condition Code		Description	Class of Condition Code
iwarp_cc77.h	iwarp_cc.h		
ALWAYS	IW_ALWAYS	always true	Miscellaneous
BNDPRT	IW_BINDPROT	BIND OP PROTECTED	MODE Register
C1EXCE	IW_C1EXCEN	CLASS 1 EXCEPTION ENABLE	MODE Register
C2EXCE	IW_C2EXCEN	CLASS 2 EXCEPTION ENABLE	MODE Register
C3EXCE	IW_C3EXCEN	CLASS 3 EXCEPTION ENABLE	MODE Register
C4EXCE	IW_C4EXCEN	CLASS 4 EXCEPTION ENABLE	MODE Register
CACHEF	IW_CACHEFAIL	CACHE FAILURE	STATE Register
DBGEVT	IW_DEBUGEVT	DEBUG BREAK	EVENT Register
EXTEVT	IW_EXTEVT	EXTERNAL EVENT	EVENT Register
FAILED	IW_FAILED	FAIL bit set in CONFIG	Miscellaneous
G0R1W	IW_G0R_1WRD	G0 has 1 WORD AVAILABLE	Stream Gate Status
G0R2W	IW_G0R_2WRD	G0 has 2 WORDS AVAILABLE	Stream Gate Status
G0RCCM	IW_G0R_CCM	CLOSE CONMSG	Stream Data Tags
G0RCC	IW_G0R_CCON	CLOSE CONNECTION	Stream Data Tags

Table C-1: Asynchronous Condition Codes (cont'd)

G0RCM	IW_G0R_CMSG	CLOSE MESSAGE	Stream Data Tags
G0RMRK	IW_G0R_MARK	SOFTWARE MARK	Stream Data Tags
G0ROCM	IW_G0R_OCM	OPEN CONMSG	Stream Data Tags
G0ROC	IW_G0R_OCON	OPEN CONNECTION	Stream Data Tags
G0ROM	IW_G0R_OMSG	OPEN MESSAGE	Stream Data Tags
G0RTAG	IW_G0R_TAGS	ANY TAG	Stream Data Tags
G0RVAL	IW_G0R_VAL	G0R has a VALID BINDING	Stream Gate Status
G0W1H	IW_G0W_1HOL	G0W has 1 HOLE AVAILABLE	Stream Gate Status
G0W2H	IW_G0W_2HOL	G0W has 2 HOLES AVAILABLE	Stream Gate Status
G0WVAL	IW_G0W_VAL	G0W has a VALID BINDING	Stream Gate Status
G1R1W	IW_G1R_1WRD	G1R has 1 WORD AVAILABLE	Stream Gate Status
G1R2W	IW_G1R_2WRD	G1R has 2 WORDS AVAILABLE	Stream Gate Status
G1RCCM	IW_G1R_CCM	CLOSE CONMSG	Stream Data Tags
G1RCC	IW_G1R_CCON	CLOSE CONNECTION	Stream Data Tags
G1RCM	IW_G1R_CMSG	CLOSE MESSAGE	Stream Data Tags
G1RMRK	IW_G1R_MARK	SOFTWARE MARK	Stream Data Tags
G1ROCM	IW_G1R_OCM	OPEN CONMSG	Stream Data Tags
G1ROC	IW_G1R_OCON	OPEN CONNECTION	Stream Data Tags
G1ROM	IW_G1R_OMSG	OPEN MESSAGE	Stream Data Tags
G1RTAG	IW_G1R_TAGS	ANY TAG	Stream Data Tags
G1RVAL	IW_G1R_VAL	G1R has a VALID BINDING	Stream Gate Status
G1W1H	IW_G1W_1HOL	G1W has 1 HOLE AVAILABLE	Stream Gate Status
G1W2H	IW_G1W_2HOL	G1W has 2 HOLES AVAILABLE	Stream Gate Status
G1WVAL	IW_G1W_VAL	G1W has a VALID BINDING	Stream Gate Status
GXRCCM	IW_GXR_CCM	CLOSE CONMSG	Stream Data Tags
GXRCC	IW_GXR_CCON	CLOSE CONNECTION	Stream Data Tags
GXRCM	IW_GXR_CMSG	CLOSE MESSAGE	Stream Data Tags
GXRMRK	IW_GXR_MARK	SOFTWARE MARK	Stream Data Tags
GXROCM	IW_GXR_OCM	OPEN CONMSG	Stream Data Tags
GXROC	IW_GXR_OCON	OPEN CONNECTION	Stream Data Tags
GXROM	IW_GXR_OMSG	OPEN MESSAGE	Stream Data Tags
GXRTAG	IW_GXR_TAGS	ANY TAG	Stream Data Tags
RPIRES	IW_ILU_RPIRES	RPI RESERVED	ILU Flag Register
IUSR0	IW_ILU_USR0	USER FLAG 0	ILU Flag Register
IUSR1	IW_ILU_USR1	USER FLAG 1	ILU Flag Register
IUSR2	IW_ILU_USR2	USER FLAG 2	ILU Flag Register
IUSR3	IW_ILU_USR3	USER FLAG 3	ILU Flag Register
INDPND	IW_INDPEND	INDIRECTION PENDING	STATE Register
INDRGV	IW_INDRGVAL	INDIRECT REGS VALID	STATE Register
INSTTO	IW_INSTRTO	INSTR TIMEOIUT	EVENT Register
LM1PND	IW_LM1PEND	LM OP1 INCOMPLETE	STATE Register
LM2PND	IW_LM2PEND	LM OP2 INCOMPLETE	STATE Register
LMPRTY	IW_LMPARITY	LM PARITY FAIL	EVENT Register
LMRNG	IW_LMRANGE	LM ADDRESS OVERFLOW	EVENT Register
LMTO	IW_LMTO	LM TIMEOUT	EVENT Register
MATCHA	IW_MATCHA	MATCHED ARRIVAL	EVENT Register
OPRND	IW_OPRND	ILLEGAL OPERAND	EVENT Register
PARITY	IW_PARITY	PARITY FAIL	EVENT Register
PROTEN	IW_PROTEN	PROTECTION ENABLE	MODE Register
PRTEVT	IW_PROTEVT	PROTECTION VIOLATION	EVENT Register
FPRND0	IW_FPRND0	FP ROUND bit 0	MODE Register
FPRND1	IW_FPRND1	FP ROUND bit 1	MODE Register
RQMT	IW_RQMT	RECIP QUEUE EMPTY	EVENT Register
SPINTO	IW_SPINTO	SPIN TIMEOUT	EVENT Register
SPLAB	IW_SPLFAIL	SPOOL ABORT	STATE Register

Table C-1: Asynchronous Condition Codes (cont'd)

SPOOL0	IW_SPOOL0	SPOOL 0 EVENT	EVENT Register
SPOOL1	IW_SPOOL1	SPOOL 1 EVENT	EVENT Register
SPOOL2	IW_SPOOL2	SPOOL 2 EVENT	EVENT Register
SPOOL3	IW_SPOOL3	SPOOL 3 EVENT	EVENT Register
SPOOL4	IW_SPOOL4	SPOOL 4 EVENT	EVENT Register
SPOOL5	IW_SPOOL5	SPOOL 5 EVENT	EVENT Register
SPOOL6	IW_SPOOL6	SPOOL 6 EVENT	EVENT Register
SPOOL7	IW_SPOOL7	SPOOL 7 EVENT	EVENT Register
SPOOLE	IW_SPOOLEN	SPOOLING ENABLE	MODE Register
SRUEVT	IW_SRUEVT	SRU EVENT	EVENT Register
STKHI	IW_STKHI	STACK HIGH LIMIT	EVENT Register
STKLO	IW_STKLO	STACK LOW LIMIT	EVENT Register
STOPPD	IW_STOPPED	EXPRESS STOPPED	EVENT Register
USRMOD	IW_USERMODE	USER MODE	MODE Register
USRTO0	IW_USRTO0	USER TIMEOUT #0	EVENT Register
USRTO1	IW_USRTO1	USER TIMEOUT #1	EVENT Register
XQMT	IW_XQMT	BOTH QUEUES EMPTY	EVENT Register

Unsafe/Synchronous Condition Codes

Table C-2 provides a complete list of the condition codes associated with synchronous conditions. All these codes are considered unsafe for use in _COND and _CONDCALL constructs. Users needing to test such conditions should build assembly code which both generates and tests the condition code. Such assembly code could be assembled separately and linked in or coded into an asm-inline. The condition code names listed here are defined by include "f77/iwarp_math_cc.h" or include "f77/iwarp_cc77.h" .

Table C-2: Synchronous Condition Codes

Condition Code		Description	Class of Condition Code
iwarp_math_cc77.h	iwarp_math_cc.h		
ACONC	IW_FPA_CONC	CONVERSION ERROR	FPA Flag Register
ADIV0	IW_FPA_DIV0	DIVIDE BY ZERO	FPA Flag Register
FPADNA	IW_FPA_DNA	DENORM A INPUT	FPA Flag Register
ADNALS	IW_FPA_DNALS	LOGB/SCALE DNA	FPA Flag Register
FPADNB	IW_FPA_DNB	DENORM B INPUT	FPA Flag Register
AGUARD	IW_FPA_GUARD	GUARD BIT	FPA Flag Register
AINEX	IW_FPA_INEX	INEXACT RESULT	FPA Flag Register
FPAINF	IW_FPA_INF	INFINITY	FPA Flag Register
FPAINV	IW_FPA_INVAL	INVALID OP	FPA Flag Register
FPANAN	IW_FPA_NAN	ANY NaN	FPA Flag Register
FPANEG	IW_FPA_NEG	NEGATIVE	FPA Flag Register
AOVFL	IW_FPA_OVFL	OVERFLOW	FPA Flag Register
FPAPOS	IW_FPA_POS	POSITIVE	FPA Flag Register
AQNAN	IW_FPA_QNAN	QUIET NaN	FPA Flag Register
ASNAN	IW_FPA_SNAN	SIGNALING NaN	FPA Flag Register
ASTKY	IW_FPA_STKY	STICKY BIT	FPA Flag Register
AUNFL	IW_FPA_UNFL	UNDERFLOW	FPA Flag Register
AWRAPA	IW_FPA_WRAPA	WRAPPED A INPUT	FPA Flag Register
AWRAPR	IW_FPA_WRAPR	WRAPPED RESULT	FPA Flag Register
FPAZ	IW_FPA_ZERO	ZERO	FPA Flag Register

Table C-2: Synchronous Condition Codes (cont'd)

FPAERR	IW_FPAERR	FPA ERROR	EVENT Register
FPAEXT	IW_FPAEXT	FPA EXTENSION	EVENT Register
FPAPND	IW_FPAPEND	FPA OP INCOMPLETE	STATE Register
MDIV0	IW_FPM_DIV0	DIVIDE BY ZERO	FPM Flag Register
FPMDNA	IW_FPM_DNA	DENORM A INPUT	FPM Flag Register
FPMDNB	IW_FPM_DNB	DENORM B INPUT	FPM Flag Register
MFMTE	IW_FPM_FMTE	FORMAT ERROR	FPM Flag Register
MGUARD	IW_FPM_GUARD	GUARD BIT	FPM Flag Register
MINEX	IW_FPM_INEX	INEXACT RESULT	FPM Flag Register
FPMINF	IW_FPM_INF	INFINITY	FPM Flag Register
MINTOV	IW_FPM_INTOV	INTEGER OVERFLOW	FPM Flag Register
FPMINV	IW_FPM_INVAL	INVALID OP	FPM Flag Register
FPMNAN	IW_FPM_NAN	ANY NaN	FPM Flag Register
FPMNEG	IW_FPM_NEG	NEGATIVE	FPM Flag Register
MOVFL	IW_FPM_OVFL	OVERFLOW	FPM Flag Register
FPMPOS	IW_FPM_POS	POSITIVE	FPM Flag Register
MQNAN	IW_FPM_QNAN	QUIET NaN	FPM Flag Register
MSTKY	IW_FPM_STKY	STICKY BIT	FPM Flag Register
MUNFL	IW_FPM_UNFL	UNDERFLOW	FPM Flag Register
MWRAPA	IW_FPM_WRAPA	WRAPPED A INPUT	FPM Flag Register
MWRAPB	IW_FPM_WRAPB	WRAPPED B INPUT	FPM Flag Register
MWRAPR	IW_FPM_WRAPR	WRAPPED RESULT	FPM Flag Register
FPMZ	IW_FPM_ZERO	ZERO	FPM Flag Register
FPMERR	IW_FPMERR	FPM ERROR	EVENT Register
FPMEXT	IW_FPMEXT	FPM EXTENSION	EVENT Register
FPMPND	IW_FPMPEND	FPM OP INCOMPLETE	STATE Register
ILUCY	IW_ILU_CY	CARRY	ILU Flag Register
ILPZ	IW_ILU_LPZERO	LOOP COUNT ZERO	ILU Flag Register
ILULSB	IW_ILU_LSB	LEAST SIGNIFICANT BIT	ILU Flag Register
ILUMSB	IW_ILU_MSB	MOST SIGNIFICANT BIT	ILU Flag Register
ILUNEG	IW_ILU_NEG	NEGATIVE	ILU Flag Register
IOVFL	IW_ILU_OVFL	OVERFLOW	ILU Flag Register
IOVFLS	IW_ILU_OVFLS	STICKY OVERFLOW	ILU Flag Register
ILUPOS	IW_ILU_POS	POSTIVE	ILU Flag Register
IZERO	IW_ILU_ZERO	ZERO	ILU Flag Register

Control Space Registers

Table C-3 lists the control space register definitions. There is a defined constant CSR_INC (CSRINC in regnums77.h) that defines the stride between adjacent CS registers starting with the same name and ending with an increasing sequence of single digits, such as CSR_SPOOLB0, CSR_SPOOLB1. The register names listed here are defined by include ‘‘f77/regnums.h” or include ‘‘f77/regnums77.h” .

Table C-3: Control Space Register Definitions

CSR Name		Description
regnums77.h	regnums.h	
		Status/Modes
CONST	CSR_CONSTAT	Configuration/Status Register
EVENT	CSR_EVENT	Event Register
EVENTC	CSR_EVENTC	Event Clear
EVENTR	CSR_EVENTR	Event Report Enables
MODE	CSR_MODE	Program Modes
GLOBEC	CSR_GLOBENC	Global Enable Clear
GLOBES	CSR_GLOBENS	Global Enable Set
STATE	CSR_STATE	Program State
TIMER0	CSR_TIMER0	GP Time #1: Read/Write
TIM0E	CSR_TIMER0E	GP Timer #0: Enable
TIM0D	CSR_TIMER0D	GP Timer #0: Disable
TIMER1	CSR_TIMER1	GP Timer #1: Read/Write
TIM1E	CSR_TIMER1E	GP Timer #1: Enable
TIM1D	CSR_TIMER1D	GP Timer #1: Disable
STIMER	CSR_STIMER	State Timer: Read/Write
STIME	CSR_STIMERE	State Timer: Enable
STIMD	CSR_STIMERD	State Timer: Disable
		Local Memory Interface
CLRCAC	CSR_CLRCAC	Clear Cache
LMPROT	CSR_LMPROT	Protection Configuration
LMRTH	CSR_LMRTH	LMR Transfer, high word
LMRTL	CSR_LMRTL	LMR Transfer, low word
LMSIZE	CSR_LMSIZE	Memory Size
MAR	CSR_MAR	Memory Address
MDRH	CSR_MDRH	Memory Data, high word
MDRL	CSR_MDRL	Memory Data, low word
PC	CSR_PC	Program Counter
PCFAIL	CSR_PCFAIL	PC on abort
RFDEST	CSR_RFDEST	RF Destination
ROMMSK	CSR_ROMMSK	ROM Patch mask
SCBASE	CSR_SCBASE	Supervisor Call Table
SP	CSR_SP	Stack Pointer
SPLO	CSR_SPLO	Stack Lower Limit
SPHI	CSR_SPHI	Stack Upper Limit
XBASE	CSR_XBASE	Exception Base Address
		General Instruction Support
ILUFLG	CSR_ILUFLG	ILU Flags
LOOPC	CSR_LOOPC	Loop Count/Condition
LOOPA	CSR_LOOPA	Loop Address/Enables
NEXTRA	CSR_NEXTRA	Next A Operand RF Address
NEXTRB	CSR_NEXTRB	Next B/R Operand RF Address

Table C-3: Control Space Register Definitions (cont'd)

		Floating Point Units
FPAFLG	CSR_FPAFLG	FPA Flags
FPAEVT	CSR_FPAEVT	FPA Event Enables
FAINAL	CSR_FAINAL	FPA A-Input, low word
FAINAH	CSR_FAINAH	FPA A-Input, high word
FAINBL	CSR_FAINBL	FPA B-Input, low word
FAINBH	CSR_FAINBH	FPA B-Input, high word
FAOUTL	CSR_FAOUTL	FPA Output, low word
FAOUTH	CSR_FAOUTH	FPA Output, low word
FAOPER	CSR_FAOPER	FPA Operation
FPMFLG	CSR_FPMFLG	FPM Flags
FPMEVT	CSR_FPMEVT	FPM Event Enables
FMINML	CSR_FMINML	FPM A-Input, low word
FMINMH	CSR_FMINMH	FPM A-Input, high word
FMINNL	CSR_FMINNL	FPM B-Input, low word
FMINNH	CSR_FMINNH	FPM B-Input, high word
FMOUTL	CSR_FMOUTL	FPM Output, low word
FMOUTH	CSR_FMOUTH	FPM Output, low word
FMOPER	CSR_FMOPER	FPM Operation
		Streaming/Spooling Controls
STRMB	CSR_STREAMB	Stream Bindings
SPLA0	CSR_SPOOLA0	Spool RF Binding 0
SPLA1	CSR_SPOOLA1	Spool RF Binding 1
SPLA2	CSR_SPOOLA2	Spool RF Binding 2
SPLA3	CSR_SPOOLA3	Spool RF Binding 3
SPLA4	CSR_SPOOLA4	Spool RF Binding 4
SPLA5	CSR_SPOOLA5	Spool RF Binding 5
SPLA6	CSR_SPOOLA6	Spool RF Binding 6
SPLA7	CSR_SPOOLA7	Spool RF Binding 7
SPLB0	CSR_SPOOLB0	Spool Binding 0
SPLB1	CSR_SPOOLB1	Spool Binding 1
SPLB2	CSR_SPOOLB2	Spool Binding 2
SPLB3	CSR_SPOOLB3	Spool Binding 3
SPLB4	CSR_SPOOLB4	Spool Binding 4
SPLB5	CSR_SPOOLB5	Spool Binding 5
SPLB6	CSR_SPOOLB6	Spool Binding 6
SPLB7	CSR_SPOOLB7	Spool Binding 7
		Pathway Unit Controls
AMCID0	CSR_AMCID0	AMC Address 0
AMCID1	CSR_AMCID1	AMC Address 1
AMCID2	CSR_AMCID2	AMC Address 2
AMCID3	CSR_AMCID3	AMC Address 3
AMCMK0	CSR_AMCMK0	AMC Mask 0
AMCMK1	CSR_AMCMK1	AMC Mask 1
AMCMK2	CSR_AMCMK2	AMC Mask 2
AMCMK3	CSR_AMCMK3	AMC Mask 3
MATCH	CSR_MATCH	Matched Arrival
MATCHR	CSR_MATCHR	MATCH Report Status
NIMSG	CSR_NIMSG	Not In Message
PBDLAY	CSR_PBDELAY	Pin Buffer Delay Control
PBPREG	CSR_PBPREG	Pin Buffer Parity
QMT	CSR_QMT	Outbound Queue Empty Status
RMT	CSR_RMT	Recipient Queue Empty Status

Table C-3: Control Space Register Definitions (cont'd)

RMTR	CSR_RMTR	RMT Report Enable
RPI0	CSR_RPI0	Recipient PCT Index 0
RPI1	CSR_RPI1	Recipient PCT Index 1
RPI2	CSR_RPI2	Recipient PCT Index 2
RPI3	CSR_RPI3	Recipient PCT Index 3
RPI4	CSR_RPI4	Recipient PCT Index 4
RPI5	CSR_RPI5	Recipient PCT Index 5
RPI6	CSR_RPI6	Recipient PCT Index 6
RPI7	CSR_RPI7	Recipient PCT Index 7
RPI8	CSR_RPI8	Recipient PCT Index 8
RPI9	CSR_RPI9	Recipient PCT Index 9
RPI10	CSR_RPI10	Recipient PCT Index 10
RPI11	CSR_RPI11	Recipient PCT Index 11
RPI12	CSR_RPI12	Recipient PCT Index 12
RPI13	CSR_RPI13	Recipient PCT Index 13
RPI14	CSR_RPI14	Recipient PCT Index 14
RPI15	CSR_RPI15	Recipient PCT Index 15
RPI16	CSR_RPI16	Recipient PCT Index 16
RPI17	CSR_RPI17	Recipient PCT Index 17
RPI18	CSR_RPI18	Recipient PCT Index 18
RPI19	CSR_RPI19	Recipient PCT Index 19
STOP	CSR_STOP	Express Stopped Status
STOPR	CSR_STOPR	STOP Report Enable
STOPSM	CSR_STOPSM	Stop on Software Mark
STOPOM	CSR_STOPOM	Stop on Open Message
STOPCM	CSR_STOPCM	Stop on Close Message
STOPCC	CSR_STOPCC	Stop on Close Connection
STOPNM	CSR_STOPNM	Stop on Not In Message
XMTR	CSR_XMTR	XMT Report Enable
		Debugging Aids
DEBUG	CSR_DEBUG	Debug Break Flags
DEBUGR	CSR_DEBUGR	Debug Break Report Enables
DEBUGC	CSR_DEBUGC	Debug Break Report Clear
IA1BP	CSR_IA1BP	Instruction Address 1 Break Point
IA2BP	CSR_IA2BP	Instruction Address 2 Break Point
DA1BP	CSR_DA1BP	Data Address 1 Break Point
DA2BP	CSR_DA2BP	Data Address 2 Break Point
RFBP	CSR_RFBP	RF Write Break Point
		System Reporting Unit
SEND	CSR_SEND	SRU Send Data Register
AUTO	CSR_AUTO	SRU Auto-Send Data Register
RECEIV	CSR_RECEIV	SRU Receive Data Register
SRSTAT	CSR_SRSTAT	SRU Status Register
SRCNTL	CSR_SRCNTL	SRU Control Register
SRCLK	CSR_SRCLK	SRU Clock Register
SRSAVE	CSR_SRSAVE	SRU State Save Register
		Test Registers - ILU
STIMC	CSR_STIMERC	State Timer Count
STIMM0	CSR_STIMERM0	STIMER Test Mode Off
STIMM1	CSR_STIMERM1	STIMER Test Mode On
STIMSC	CSR_STIMERSC	STIMER Sync Clock
STIMST	CSR_STIMERST	STIMER Reset

Table C-3: Control Space Register Definitions (cont'd)

		Test Registers - PSU
CAM0	CSR_CAM0	CAM Register 0
CAM1	CSR_CAM1	CAM Register 1
CAM2	CSR_CAM2	CAM Register 2
CAM3	CSR_CAM3	CAM Register 3
PSUDAT	CSR_PSUDAT	Data Register
UIAR	CSR_UIAR	Advanced uInstruction Bus
UATEST	CSR_UATEST	Test Address
		Test Registers - FPU
FPATST	CSR_FPATEST	FPA Test Control
FPMTST	CSR_FPMTEST	FPM Test Control
DSRTL	CSR_DSROOTL	FPM root/divisor, low word
DSRTH	CSR_DSROOTH	FPM root/divisor,. high word
QUORDL	CSR_QUORADL	FPM quotient/radicand, low word
QUORDH	CSR_QUORADH	FPM quotient/radicand, high word
PREML	CSR_PREML	FPM partial remainder, low word
PREMH	CSR_PREMH	FPM partial remainder, high word
		Test Registers - PWU
DIN	CSR_DIN	Direction In
DOUT0	CSR_DOUT0	Direction Out 0
DOUT1	CSR_DOUT1	Direction Out 1
FLUSH	CSR_FLUSH	Flush
HLD	CSR_HLD	Hold Scheduling
LPTC0	CSR_LPTC0	RPI CAM Bits 0
LPTC1	CSR_LPTC1	RPI CAM Bits 1
LPTC2	CSR_LPTC2	RPI CAM Bits 2
LPTC3	CSR_LPTC3	RPI CAM Bits 3
LPTC4	CSR_LPTC4	RPI CAM Bits 4
MSKTOQ	CSR_MSKTOQ	Mask Top of Queue
NBFCT0	CSR_NBFCNT0	Recipient Buffer Count 0
NBFCT1	CSR_NBFCNT1	Recipient Buffer Count 1
NBFCT2	CSR_NBFCNT2	Recipient Buffer Count 2
NBFCT3	CSR_NBFCNT3	Recipient Buffer Count 3
PBCNTL	CSR_PBCONTL	Pin Buffer Control
PIN	CSR_PIN	Pathway In
POOL	CSR_POOL	Pool
POUT0	CSR_POUT0	Pathway Out 0
POUT1	CSR_POUT1	Pathway Out 1
SDQ	CSR_SDQ	Send DQ Tags
SELD2	CSR_SELD2	Spool Debug Register 2
SELD4	CSR_SELD4	Spool Debug Register 4
SELD7	CSR_SELD7	Spool Debug Register 7
SEN	CSR_SEN	Scheduling Enabled
SP1W	CSR_SP1W	Spool 1 Word
SPN0	CSR_SPN0	Spool Number 0
SPN1	CSR_SPN1	Spool Number 1
SPN2	CSR_SPN2	Spool Number 2
SPV	CSR_SPV	Spool Valid
SPNLST	CSR_SPLNST	Spool Next Start Location
STP	CSR_STP	Single Step Scheduler
TAS0	CSR_TAS0	RPI Assigned 0
TAS1	CSR_TAS1	RPI Assigned 1

Table C-3: Control Space Register Definitions (cont'd)

TAS2	CSR_TAS2	RPI Assigned 2
TAS3	CSR_TAS3	RPI Assigned 3
TAS4	CSR_TAS4	RPI Assigned 4
TAS5	CSR_TAS5	RPI Assigned 5
TAS6	CSR_TAS6	RPI Assigned 6
TAS7	CSR_TAS7	RPI Assigned 7
TAS8	CSR_TAS8	RPI Assigned 8
TAS9	CSR_TAS9	RPI Assigned 9
TAS10	CSR_TAS10	RPI Assigned 10
TAS11	CSR_TAS11	RPI Assigned 11
TAS12	CSR_TAS12	RPI Assigned 12
TAS13	CSR_TAS13	RPI Assigned 13
TAS14	CSR_TAS14	RPI Assigned 14
TAS15	CSR_TAS15	RPI Assigned 15
TAS16	CSR_TAS16	RPI Assigned 16
TAS17	CSR_TAS17	RPI Assigned 17
TAS18	CSR_TAS18	RPI Assigned 18
TAS19	CSR_TAS19	RPI Assigned 19
TME	CSR_TME	Tag Match Enable
TOLM	CSR_TOLM	Spool To LM
XL0REG	CSR_XL0REG	XLeft Pool 0 RPI
XL1REG	CSR_XL1REG	XLeft Pool 1 RPI
XR0REG	CSR_XR0REG	XRight Pool 0 RPI
XR1REG	CSR_XR1REG	XRight Pool 1 RPI
YL0REG	CSR_YL0REG	YLeft Pool 0 RPI
YL1REG	CSR_YL1REG	YLeft Pool 1 RPI
YR0REG	CSR_YR0REG	YRight Pool 0 RPI
YR1REG	CSR_YR1REG	YRight Pool 1 RPI

Register File Definitions

Table C-4 lists the register file definitions. There is a defined constant REG_INC (REGINC in regnums77.h) that defines the stride between adjacent registers starting with the same name and ending with an increasing sequence of numbers, such as REG_SC0 and REG_SC1. The register names listed here are defined by include ''f77/regnums.h'' or include ''f77/regnums77.h'' . The names listed are for regnums.h; regnums77.h names omit the REG_ prefixes.

Table C-4: Register File Definitions

Scratch Registers

REG_SR0	REG_SR4	REG_SR8	REG_SR12
REG_SR1	REG_SR5	REG_SR9	REG_SR13
REG_SR2	REG_SR6	REG_SR10	REG_SR14
REG_SR3	REG_SR7	REG_SR11	REG_SR15

Table C-4: Register File Definitions (cont'd)

Permanent Registers

REG_PR0	REG_PR12	REG_PR24	REG_PR36	REG_PR48
REG_PR1	REG_PR13	REG_PR25	REG_PR37	REG_PR49
REG_PR2	REG_PR14	REG_PR26	REG_PR38	REG_PR50
REG_PR3	REG_PR15	REG_PR27	REG_PR39	REG_PR51
REG_PR4	REG_PR16	REG_PR28	REG_PR40	REG_PR52
REG_PR5	REG_PR17	REG_PR29	REG_PR41	REG_PR53
REG_PR6	REG_PR18	REG_PR30	REG_PR42	REG_PR54
REG_PR7	REG_PR19	REG_PR31	REG_PR43	REG_PR55
REG_PR8	REG_PR20	REG_PR33	REG_PR44	
REG_PR9	REG_PR21	REG_PR34	REG_PR45	
REG_PR10	REG_PR22	REG_PR35	REG_PR46	
REG_PR11	REG_PR23	REG_PR36	REG_PR47	

Argument Registers

REG_AR0	REG_AR4	REG_AR8	REG_AR12
REG_AR1	REG_AR5	REG_AR9	REG_AR13
REG_AR2	REG_AR6	REG_AR10	REG_AR14
REG_AR3	REG_AR7	REG_AR11	REG_AR15

Function Value Registers

REG_FV0	REG_FV1	REG_FV2	REG_FV3

Frame and Argument Block Pointer Registers

REG_FP	REG_AP

Event Handler Registers

REG_EV0	REG_EV2	REG_EV4	REG_EV6
REG_EV1	REG_EV3	REG_EV5	REG_EV7

Spooling Registers

REG_SC0	REG_SC4	REG_SC8	REG_SC12
REG_SC1	REG_SC5	REG_SC9	REG_SC13
REG_SC2	REG_SC6	REG_SC10	REG_SC14
REG_SC3	REG_SC7	REG_SC11	REG_SC15

C&A Local Memory Access Registers

REG_LMR2		REG_LMR1		REG_LMW	
REG_LM0	REG_LM1	REG_LM2	REG_LM3	REG_LM4	REG_LM5

Gate Registers

REG_GA0	REG_GA1	REG_GA2	REG_GA3