Appendix B: iWarp Include Files

The following include files can be found in /iwarp/include and can be included by using **#include** **<*name.h*>**:

a.out.h	COFF related definitions, includes the files: filehdr, aouthdr, scnhdr, reloc, linenum, syms, storclass.
aouthdr.h	COFF related definitions.
ar.h	Common archive format definitions.
asm/*.h	Asm–inline definition files, some of which are released in conjunction with the iWarp runtime system. This includes:

asm/f77intrinsics.h	definitions for numerous FORTRAN intrinsics (names vary however)
asm/gen_type.h	type definitions included by gen_asm.h
asm/gen_asm.h	iWarp common use asm–inline definitions.
asm/isib.h	iWarp Sun Interface Board definitions.
asm/iwarp_fp.h	iWarp IEEE floating-point functions as asm–inline definitions.
asm/pw_type.h	type definition included by pw_asm.h
asm/pw_asm.h	iWarp pathway control asm–inline definitions.

assert.h	Defines the assert macro.
bool.h	Defines TRUE and FALSE.
ctype.h	Character macro definitions.
errno.h	Error code definition.
fcntl.h	File control definitions.
filehdr.h	COFF related definitions.
gates.h	Definitions of *GATE0* and *GATE1* (included by stdio.h).
iwarp_cc.h	Safe (asynchronous) iWarp condition codes for **_cond** and **_condcall**.
ldfcn.h	LDFILE definitions.
limits.h	Useful value definitions.
linenum.h	*struct lineno* definitions.
malloc.h	malloc definitions.
math.h	Mathematical definitions.

memory .h	Memory function definitions.
mon.h	Profiler information definitions.
nan.h	Nan floating-point definitions.
nlist.h	Symbol table definitions.
patch.h	*struct plist* definitions.
paths.h	Definition of paths to standard areas (/iwarp/...).
regnums.h	Register name definitions for asm–inlines.
reloc.h	COFF-related definitions.
scnhdr .h	COFF-related definitions.
search.h	Tree structure definitions.
setjmp.h	setjmp and longjmp definitions.
sgs.h	Useful definitions for iWarp tools.
sgsmacros.h	Useful macros for iWarp tools.
stdio.h	Standard I/O definitions (includes gates.h and stdio2.h). This file is needed in almost all iWarp application programs.
storclass.h	Storage classes.
string.h	String function prototypes.
strings.h	String function prototypes.
syms.h	COFF-related definitions.
sys/ieeefp.h	iWarp Floating point arithmetic interfaces: (1) Rounding Control (2) Exception Control (3) Exception Handling (4) Utility Macros (5) Full Exception Environment Control
sys/iwarp_math_cc.h	Unsafe (synchronous) iWarp condition codes for **_cond** and **_condcall** .
time.h	COFF-related definitions.
tv.h	TV definitions.
values.h	Definitions of MAXDOUBLE, MAXFLOAT, MINDOUBLE, MINFLOA T and other useful values.
varargs.h	varargs definitions.

Safe (Asynchronous) Condition Codes

Table B-1 contains a complete list of the condition codes associated with asynchronous conditions. All of these codes are safe for use in _cond and _condcall constructs. The condition code names listed in Table B-1 are defined in <iwarp_cc.h>.

Table B-1: Safe Condition Codes

Condition Code	Description	Class of Condition Code
IW_ALWAYS	always true	Miscellaneous
IW_BINDPROT	BIND OP PROTECTED	MODE Register
IW_C1EXCEN	CLASS 1 EXCEPTION ENABLE	MODE Register
IW_C2EXCEN	CLASS 2 EXCEPTION ENABLE	MODE Register
IW_C3EXCEN	CLASS 3 EXCEPTION ENABLE	MODE Register
IW_C4EXCEN	CLASS 4 EXCEPTION ENABLE	MODE Register
IW_DEBUGEVT	DEBUG BREAK	EVENT Register
IW_EXTEVT	EXTERNAL EVENT	EVENT Register
IW_G0R_1WRD	G0 has 1 WORD AVAILABLE	Stream Gate Status
IW_G0R_2WRD	G0 has 2 WORDS AVAILABLE	Stream Gate Status
IW_G0R_CCM	CLOSE CONMSG	Stream Data Tags
IW_G0R_CCON	CLOSE CONNECTION	Stream Data Tags
IW_G0R_CMSG	CLOSE MESSAGE	Stream Data Tags
IW_G0R_MARK	SOFTWARE MARK	Stream Data Tags
IW_G0R_OCM	OPEN CONMSG	Stream Data Tags
IW_G0R_OCON	OPEN CONNECTION	Stream Data Tags
IW_G0R_OMSG	OPEN MESSAGE	Stream Data Tags
IW_G0R_TAGS	ANY TAG	Stream Data Tags
IW_G0R_VAL	G0R has a VALID BINDING	Stream Gate Status
IW_G0W_1HOL	G0W has 1 HOLE AVAILABLE	Stream Gate Status
IW_G0W_2HOL	G0W has 2 HOLES AVAILABLE	Stream Gate Status
IW_G0W_VAL	G0W has a VALID BINDING	Stream Gate Status
IW_G1R_1WRD	G1R has 1 WORD AVAILABLE	Stream Gate Status
IW_G1R_2WRD	G1R has 2 WORDS AVAILABLE	Stream Gate Status
IW_G1R_CCM	CLOSE CONMSG	Stream Data Tags
IW_G1R_CCON	CLOSE CONNECTION	Stream Data Tags
IW_G1R_CMSG	CLOSE MESSAG	Stream Data Tags
IW_G1R_MARK	SOFTWARE MARK	Stream Data Tags
IW_G1R_OCM	OPEN CONMSG	Stream Data Tags
IW_G1R_OCON	OPEN CONNECTION	Stream Data Tags
IW_G1R_OMSG	OPEN MESSAGE	Stream Data Tags
IW_G1R_TAGS	ANY TAG	Stream Data Tags
IW_G1R_VAL	G1R has a VALID BINDING	Stream Gate Status
IW_G1W_1HOL	G1W has 1 HOLE AVAILABLE	Stream Gate Status
IW_G1W_2HOL	G1W has 2 HOLES AVAILABLE	Stream Gate Status
IW_G1W_VAL	G1W has a VALID BINDING	Stream Gate Status
IW_GLOBENB	Global Event enable	MODE Register
IW_GXR_CCM	CLOSE CONMSG	Stream Data Tags
IW_GXR_CCON	CLOSE CONNECTION	Stream Data Tags
IW_GXR_CMSG	CLOSE MESSAGE	Stream Data Tags
IW_GXR_MARK	SOFTWARE MARK	Stream Data Tags
IW_GXR_OCM	OPEN CONMSG	Stream Data Tags
IW_GXR_OCON	OPEN CONNECTION	Stream Data Tags
IW_GXR_OMSG	OPEN MESSAGE	Stream Data Tags
IW_GXR_TAGS	ANY TAG	Stream Data Tags
IW_ILU_RPIRES	RPI RESERVED	ILU Flag Register
IW_ILU_USR0	USER FLAG 0	ILU Flag Register
IW_ILU_USR1	USER FLAG 1	ILU Flag Register
IW_ILU_USR2	USER FLAG 2	ILU Flag Register
IW_ILU_USR3	USER FLAG 3	ILU Flag Register
IW_INDPEND	INDIRECTION PENDING	MODE Register
IW_INDRGVAL	INDIRECT REGS VALID	MODE Register

Table B-1: Safe Condition Codes, cont'd

Condition Code	Description	Class of Condition Code
IW_INSTRTO	INSTR TIMEOIUT	EVENT Register
IW_LM1PEND	LM OP1 INCOMPLETE	STATERegister
IW_LM2PEND	LM OP2 INCOMPLETE	STATE Register
IW_LMPARITY	LM PARITY FAIL	EVENT Register
IW_LMRANGE	LM ADDRESS OVERFLOW	EVENT Register
IW_LMTO	LM TIMEOUT	EVENT Register
IW_MATCHA	MATCHED ARRIVAL	EVENT Register
IW_OPRND	ILLEGAL OPERAND	EVENT Register
IW_PARITY	PARITY FAIL	EVENT Register
IW_PROTEN	PROTECTION ENABLE	MODE Register
IW_PROTEVT	PROTECTION VIOLATION	EVENT Register
IW_FPRND0	FP ROUND bit 0	MODE Register
IW_FPRND1	FP ROUND bit 1	MODE Register
IW_RQMT	RECIP QUEUE EMPTY	EVENT Register
IW_SPINTO	SPIN TIMEOUT	EVENT Register
IW_SPOOL0	SPOOL 0 EVENT	EVENT Register
IW_SPOOL1	SPOOL 1 EVENT	EVENT Register
IW_SPOOL2	SPOOL 2 EVENT	EVENT Register
IW_SPOOL3	SPOOL 3 EVENT	EVENT Register
IW_SPOOL4	SPOOL 4 EVENT	EVENT Register
IW_SPOOL5	SPOOL 5 EVENT	EVENT Register
IW_SPOOL6	SPOOL 6 EVENT	EVENT Register
IW_SPOOL7	SPOOL 7 EVENT	EVENT Register
IW_SPOOLEN	SPOOLING ENABLE	MODE Register
IW_SPOVFL	STACK OVERFLOW	EVENT Register
IW_SPUNFL	STACK UNDERFLOW	EVENT Register
IW_SRUEVT	SRU EVENT	EVENT Register
IW_STOPPED	EXPRESS STOPPED	EVENT Register
IW_USERMODE	USER MODE	MODE Register
IW_USRTO0	USER TIMEOUT #0	EVENT Register
IW_USRTO1	USER TIMEOUT #1	EVENT Register
IW_XQMT	BOTH QUEUES EMPTY	EVENT Register

Unsafe (Synchronous) Condition Codes

Table B-2 contains a complete list of the condition codes which are associated with synchronous conditions. All these codes are unsafe for use in _cond and _condcall constructs. If you need to test such conditions, you should build assembly code which both generates and tests the condition code. Such assembly code could be assembled separately and linked in, or it could be coded into an asm–inline. The condition code names listed in Table B-2 are defined in <sys/iwarp_math_cc.h>.

Table B-2: Unsafe Condition Codes

Condition Code	Description	Class of Condition Code
IW_FPA_CONC	CONVERSION ERROR	FPA Flag Register
IW_FPA_DIV0	DIVIDE BY ZERO	FPA Flag Register
IW_FPA_DNA	DENORM A INPUT	FPA Flag Register
IW_FPA_DNALS	LOGB/SCALE DNA	FPA Flag Register
IW_FPA_DNB	DENORM B INPUT	FPA Flag Register
IW_FPA_GUARD	GUARD BIT	FPA Flag Register
IW_FPA_INEX	INEXACT RESULT	FPA Flag Register
IW_FPA_INF	INFINITY	FPA Flag Register
IW_FPA_INVAL	INVALID OP	FPA Flag Register
IW_FPA_NAN	ANY NaN	FPA Flag Register
IW_FPA_NEG	NEGATIVE	FPA Flag Register
IW_FPA_OVFL	OVERFLOW	FPA Flag Register
IW_FPA_POS	POSITIVE	FPA Flag Register
IW_FPA_QNAN	QUIET NaN	FPA Flag Register
IW_FPA_SNAN	SIGNALING NaN	FPA Flag Register
IW_FPA_STKY	STICKY BIT	FPA Flag Register
IW_FPA_UNFL	UNDERFLOW	FPA Flag Register
IW_FPA_WRAPA	WRAPPED A INPUT	FPA Flag Register
IW_FPA_WRAPR	WRAPPED RESULT	FPA Flag Register
IW_FPA_ZERO	ZERO	FPA Flag Register
IW_FPAERR	FPA ERROR	EVENT Register
IW_FPAEXT	FPA EXTENSION	EVENT Register
IW_FPM_DIV0	DIVIDE BY ZERO	FPM Flag Register
IW_FPM_DNA	DENORM A INPUT	FPM Flag Register
IW_FPM_DNB	DENORM B INPUT	FPM Flag Register
IW_FPM_FMTE	FORMAT ERROR	FPM Flag Register
IW_FPM_GUARD	GUARD BIT	FPM Flag Register
IW_FPM_INEX	INEXACT RESULT	FPM Flag Register
IW_FPM_INF	INFINITY	FPM Flag Register
IW_FPM_INTOV	INTEGER OVERFLOW	FPM Flag Register
IW_FPM_INVAL	INVALID OP	FPM Flag Register
IW_FPM_NAN	ANY NaN	FPM Flag Register
IW_FPM_NEG	NEGATIVE	FPM Flag Register
IW_FPM_OVFL	OVERFLOW	FPM Flag Register
IW_FPM_POS	POSITIVE	FPM Flag Register
IW_FPM_QNAN	QUIET NaN	FPM Flag Register
IW_FPM_SNAN	SIGNALING NaN	FPM Flag Register
IW_FPM_STKY	STICKY BIT	FPM Flag Register
IW_FPM_UNFL	UNDERFLOW	FPM Flag Register
IW_FPM_WRAPA	WRAPPED A INPUT	FPM Flag Register
IW_FPM_WRAPB	WRAPPED B INPUT	FPM Flag Register
IW_FPM_WRAPR	WRAPPED RESULT	FPM Flag Register
IW_FPM_ZERO	ZERO	FPM Flag Register
IW_FPMERR	FPM ERROR	EVENT Register
IW_FPMEXT	FPM EXTENSION	EVENT Register
IW_ILU_CY	CARRY	ILU Flag Register
IW_ILU_LPZERO	LOOP COUNT ZERO	ILU Flag Register
IW_ILU_LSB	LEAST SIGNIFICANT BIT	ILU Flag Register
IW_ILU_MSB	MOST SIGNIFICANT BIT	ILU Flag Register
IW_ILU_NEG	NEGATIVE	ILU Flag Register
IW_ILU_OVFL	OVERFLOW	ILU Flag Register
IW_ILU_OVFLS	STICKY OVERFLOW	ILU Flag Register
IW_ILU_POS	POSTIVE	ILU Flag Register
IW_ILU_ZERO	ZERO	ILU Flag Register

Control Space Registers

Table B-3 lists the Control Space Register definitions. These definitions are in <regnums.h>. There is a defined constant **CSR_INC** which defines the stride between adjacent CS registers that start with the same name and end with an increasing sequence of single digits, such as CSR_SPOOLB0 and CSR_SPOOLB1.

Table B-3: Control Space Register Definitions

CSR Name	Description
	Status/Modes
CSR_CONSTAT	Configuration/Status Register
CSR_EVENT	Event Register
CSR_EVENTC	Event Clear
CSR_EVENTR	Event Report Enables
CSR_GLOBENC	Global Enable Clear
CSR_GLOBENS	Global Enable Clear
CSR_MODE	Program Modes
CSR_STATE	Program State
CSR_TIMER0	GP Time #1: Read/Write
CSR_TIMER0E	GP Timer #0: Enable
CSR_TIMER0D	GP Timer #0: Disable
CSR_TIMER1	GP Timer #1: Read/Write
CSR_TIMER1E	GP Timer #1: Enable
CSR_TIMER1D	GP Timer #1: Disable
CSR_STIMER	State Timer: Read/Write
CSR_STIMERE	State Timer: Enable
CSR_STIMERD	State Timer: Disable
	Local Memory Interface
CSR_CLRCAC	Clear Cache
CSR_LMPROT	Protection Configuration
CSR_LMRTH	LMR Transfer, high word
CSR_LMRTL	LMR Transfer, low word
CSR_LMSIZE	Memory Size
CSR_MAR	Memory Address
CSR_MDRH	Memory Data, high word
CSR_MDRL	Memory Data, low word
CSR_PC	Program Counter
CSR_PCFAIL	PC on abort
CSR_RFDEST	RF Destination
CSR_ROMMSK	ROM Patch mask
CSR_SCBASE	Supervisor Call Table
CSR_SP	Stack Pointer
CSR_SPLO	Stack Lower Limit
CSR_SPHI	Stack Upper Limit
CSR_XBASE	Exception Base Address

Table B-3: Control Space Register Definitions, cont'd

CSR Name	Description
	General Instruction Support
CSR_ILUFLG	ILU Flags
CSR_LOOPC	Loop Count/Condition
CSR_LOOPA	Loop Address/Enables
CSR_NEXTRA	Next A Operand RF Address
CSR_NEXTRB	Next B/R Operand RF Address
	Floating Point Units
CSR_FPAFLG	FPA Flags
CSR_FPAEVT	FPA Event Enables
CSR_FAINAL	FPA A–Input, low word
CSR_FAINAH	FPA A–Input, high word
CSR_FAINBL	FPA B–Input, low word
CSR_FAINBH	FPA B–Input, high word
CSR_FAOUTL	FPA Output, low word
CSR_FAOUTH	FPA Output, low word
CSR_FAOPER	FPA Operation
CSR_FPMFLG	FPM Flags
CSR_FPMEVT	FPM Event Enables
CSR_FMINML	FPM A–Input, low word
CSR_FMINMH	FPM A–Input, high word
CSR_FMINNL	FPM B–Input, low word
CSR_FMINNH	FPM B–Input, high word
CSR_FMOUTL	FPM Output, low word
CSR_FMOUTH	FPM Output, low word
CSR_FMOPER	FPM Operation
	Streaming/Spooling Controls
CSR_STREAM	Stream Bindings
CSR_SPOOLA0	Spool RF Binding 0
CSR_SPOOLA1	Spool RF Binding 1
CSR_SPOOLA2	Spool RF Binding 2
CSR_SPOOLA3	Spool RF Binding 3
CSR_SPOOLA4	Spool RF Binding 4
CSR_SPOOLA5	Spool RF Binding 5
CSR_SPOOLA6	Spool RF Binding 6
CSR_SPOOLA7	Spool RF Binding 7
CSR_SPOOLB0	Spool Binding 0
CSR_SPOOLB1	Spool Binding 1
CSR_SPOOLB2	Spool Binding 2
CSR_SPOOLB3	Spool Binding 3
CSR_SPOOLB4	Spool Binding 4
CSR_SPOOLB5	Spool Binding 5
CSR_SPOOLB6	Spool Binding 6
CSR_SPOOLB7	Spool Binding 7

Table B-3: Control Space Register Definitions, cont'd

CSR Name	Description
	Pathway Unit Controls
CSR_AMCID0	AMC Address 0
CSR_AMCID1	AMC Address 1
CSR_AMCID2	AMC Address 2
CSR_AMCID3	AMC Address 3
CSR_AMCMK0	AMC Mask 0
CSR_AMCMK1	AMC Mask 1
CSR_AMCMK2	AMC Mask 2
CSR_AMCMK3	AMC Mask 3
CSR_MATCH	Matched Arrival
CSR_MATCHR	MATCH Report Status
CSR_NIMSG	Not In Message
CSR_PBPREG	Pin Buffer Parity
CSR_RMT	Recipient Queue Empty Status
CSR_RMTR	RMT Report Enable
CSR_RPI0	Recipient PCT Index 0
CSR_RPI1	Recipient PCT Index 1
CSR_RPI2	Recipient PCT Index 2
CSR_RPI3	Recipient PCT Index 3
CSR_RPI4	Recipient PCT Index 4
CSR_RPI5	Recipient PCT Index 5
CSR_RPI6	Recipient PCT Index 6
CSR_RPI7	Recipient PCT Index 7
CSR_RPI8	Recipient PCT Index 8
CSR_RPI9	Recipient PCT Index 9
CSR_RPI10	Recipient PCT Index 10
CSR_RPI11	Recipient PCT Index 11
CSR_RPI12	Recipient PCT Index 12
CSR_RPI13	Recipient PCT Index 13
CSR_RPI14	Recipient PCT Index 14
CSR_RPI15	Recipient PCT Index 15
CSR_RPI16	Recipient PCT Index 16
CSR_RPI17	Recipient PCT Index 17
CSR_RPI18	Recipient PCT Index 18
CSR_RPI19	Recipient PCT Index 19
CSR_STOP	Express Stopped Status
CSR_STOPR	STOP Report Enable
CSR_STOPSM	Stop on Software Mark
CSR_STOPOM	Stop on Open Message
CSR_STOPCM	Stop on Close Message
CSR_STOPCC	Stop on Close Connection
CSR_STOPNM	Stop on Not In Message
CSR_XMTR	XMT Report Enable

Table B-3: Control Space Register Definitions, cont'd

CSR Name	Description
	Debugging Aids
CSR_DEBUG	Debug Break Flags
CSR_DEBUGR	Debug Break Report Enables
CSR_DEBUGC	Debug Break Report Clear
CSR_IA1BP	Instruction Address 1 Break Point
CSR_IA2BP	Instruction Address 2 Break Point
CSR_DA1BP	Data Address 1 Break Point
CSR_DA2BP	Data Address 2 Break Point
CSR_RFBP	RF Write Break Point
	System Reporting Unit
CSR_SEND	SRU Send Data Register
CSR_AUTO	SRU Auto–Send Data Register
CSR_RECEIV	SRU Receive Data Register
CSR_SRSTAT	SRU Status Register
CSR_SRCNTL	SRU Control Register
CSR_SRCLK	SRU Clock Register
CSR_SRSAVE	SRU State Save Register
	Test Registers – ILU
CSR_STIMERC	State Timer Count
CSR_STIMERM0	STIMER Test Mode Off
CSR_STIMERM1	STIMER Test Mode On
CSR_STIMERSC	STIMER Sync Clock
CSR_STIMERST	STIMER Reset
	Test Registers – PSU
CSR_CAM0	CAM Register 0
CSR_CAM1	CAM Register 1
CSR_CAM2	CAM Register 2
CSR_CAM3	CAM Register 3
CSR_PSUDAT	Data Register
CSR_UIAR	Advanced uInstruction Bus
CSR_UATEST	Test Address
	Test Registers – FPU
CSR_FPATEST	FPA Test Control
CSR_FPMTEST	FPM Test Control
CSR_DSROOTL	FPM root/divisor, low word
CSR_DSROOTH	FPM root/divisor,. high word
CSR_QUORADL	FPM quotient/radicand, low word
CSR_QUORADH	FPM quotient/radicand, high word
CSR_PREML	FPM partial remainder, low word
CSR_PREMH	FPM partial remainder, high word

Table B-3: Control Space Register Definitions, cont'd

CSR Name	Description
	Test Registers – PWU
CSR_DIN	Direction In
CSR_DOUT0	Direction Out 0
CSR_DOUT1	Direction Out 1
CSR_FLUSH	Flush
CSR_HLD	Hold Scheduling
CSR_LPTC0	RPI CAM Bits 0
CSR_LPTC1	RPI CAM Bits 1
CSR_LPTC2	RPI CAM Bits 2
CSR_LPTC3	RPI CAM Bits 3
CSR_LPTC4	RPI CAM Bits 4
CSR_MSKTOQ	Mask Top of Queue
CSR_NBFCNT0	Recipient Buffer Count 0
CSR_NBFCNT1	Recipient Buffer Count 1
CSR_NBFCNT2	Recipient Buffer Count 2
CSR_NBFCNT3	Recipient Buffer Count 3
CSR_PBCONTL	Pin Buffer Control
CSR_PIN	Pathway In
CSR_POOL	Pool
CSR_POUT0	Pathway Out 0
CSR_POUT1	Pathway Out 1
CSR_SDQ	Send DQ Tags
CSR_SELD2	Spool Debug Register 2
CSR_SELD4	Spool Debug Register 4
CSR_SELD7	Spool Debug Register 7
CSR_SEN	Scheduling Enabled
CSR_SP1W	Spool 1 Word
CSR_SPN0	Spool Number 0
CSR_SPN1	Spool Number 1
CSR_SPN2	Spool Number 2
CSR_SPV	Spool Valid
CSR_SPNLST	Spool Next Start Location
CSR_STP	Single Step Scheduler
CSR_TAS0	RPI Assigned 0
CSR_TAS1	RPI Assigned 1
CSR_TAS2	RPI Assigned 2
CSR_TAS3	RPI Assigned 3
CSR_TAS4	RPI Assigned 4
CSR_TAS5	RPI Assigned 5
CSR_TAS6	RPI Assigned 6
CSR_TAS7	RPI Assigned 7
CSR_TAS8	RPI Assigned 8
CSR_TAS9	RPI Assigned 9
CSR_TAS10	RPI Assigned 10
CSR_TAS11	RPI Assigned 11
CSR_TAS12	RPI Assigned 12
CSR_TAS13	RPI Assigned 13
CSR_TAS14	RPI Assigned 14
CSR_TAS15	RPI Assigned 15
CSR_TAS16	RPI Assigned 16
CSR_TAS17	RPI Assigned 17
CSR_TAS18	RPI Assigned 18
CSR_TAS19	RPI Assigned 19
CSR_TME	Tag Match Enable

CSR_TOLM	Spool To LM
CSR_XL0REG	XLeft Pool 0 RPI
CSR_XL1REG	XLeft Pool 1 RPI
CSR_XR0REG	XRight Pool 0 RPI
CSR_XR1REG	XRight Pool 1 RPI
CSR_YL0REG	YLeft Pool 0 RPI
CSR_YL1REG	YLeft Pool 1 RPI
CSR_YR0REG	YRight Pool 0 RPI
CSR_YR1REG	YRight Pool 1 RPI

Register File Definitions

Table B-4 lists the register definitions. These definitions are in <regnums.h>. There is a defined constant **REG_INC** which defines the stride between adjacent registers (which start with the same name and end with an increasing sequence of numbers) such as REG_SC0, REG_SC1, and so on.

Table B-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

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

Table B-4: Register File Definitions, cont'd

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		

Assembly Code Inline Definitions

The assembly code inline definition files are found in the /iwarp/include/asm directory . These definition files include:

`gen_type.h`	typedefs included by gen_asm.h
`gen_asm.h`	general use
`pw_type.h`	typedefs included by pw_asm.h
`pw_asm.h`	pathway control
`iwarp_fp.h`	iWarp floating-point operations

An include of iwarp_fp.h would look like:

```
#include <asm/iwarp_fp.h>
```

General typedefs – gen_type.h

The typedefs in gen_type.h are intended for inclusion by most programs using other include files in the asm directory. This file is automatically included by gen_asm.h.

General Use – gen_asm.h

The assembly code inline definitions in this file are commonly-used definitions. The typedefs.h file should be included prior to including this file. Definitions in this file include:

```
asm  void  asm_nop()
asm  void  asm_writecsreg(data, csreg)
asm  int   asm_readcsreg(csreg)
asm  void  asm_writerfreg(data, reg)
asm  int   asm_readrfreg(reg)
```

where *data* is any integer expression, *csreg* is any CSR register definition defined by #include<regnums.h> and *reg* is any regular register definition defined by #include<regnums.h>. If *csreg* or *reg* is not a constant, less efficient assembly code with register indirection will be used to allow register selection at run time. For efficiency, *csreg* and *reg* values are not checked for correct values at run time. Ensuring their correctness is the responsibility of the users program.

An **asm_nop** instantiation generates a single *nop* instruction.

General typedefs – pw_type.h

This file is automatically included by pw_asm.h and defines types useful for programs which manipulate the pathways.

Pathway Control – pw_asm.h

The assembly code inline definitions in this file are for pathway control instructions. The typedefs.h file should be included prior to including this file.

Floating-Point Math – iwarp_fp.h

The assembly code inline definitions in this file define a number of floating point routines. Including this file will cause asm–inlining to be used instead of calls to library functions. This can offer the user a way to speed up a program. The library routines corresponding to the functions defined in this file use the *same* asm–inline file to define the functions in the library.