Registers(draft)

General

COFFEE has two different register sets. The first set (SET 1) is intended to be used by application programs. The second set of registers (SET 2) is for privileged software which could be an operating system or similar. SET 2 is protected from application program. Privileged software can access both sets. There's a total of 32 registers in both sets including general purpose registers (GPRs) and special purpose registers (SPRs). In addition COFFEE has eight condition registers (CRs) which are used with conditional branches or when executing instructions conditionally. These are visible to application software as well as to privileged software. Some internal registers are not accessible (nor visible to) by the user. Those are called special function registers (SFRs) in this document. See programming guidelines for more information about register usage.

SET 1 GPRs

SET 1 has 32 identical general purpose registers R0...R31 with one exception: R31 is used as a link register by some instructions. The programmer is free to use R31 for any other purpose as long as it's special behaviour is taken into account. All general purpose registers (and the link register) are 32 bits wide.

SET 2 GPRs

SET 2 has 15 identical general purpose registers PR17...PR31 with one exception: PR31 is used as a link register by some instructions. The programmer is free to use PR31 for any other purpose as long as it's special behaviour is taken into account. All general purpose registers (and the link register) are 32 bits wide.

SET 2 SPRs

SET 2 special purpose registers are summed in the table 'SET 2 Special purpose registers'. Some of the SPRs are not 32 bit wide. When the width of some data to be written to a register exceeds the width of the register the extra bits are cut from the most significant end. When reading data from a short register the 'non existent' bits are read as zeros. Writing to a read only register is ignored. Within the following, only PSR and PSR2 are discussed. Refer to 'Interrupts and exceptions' and 'coprocessor interface' for detailed explanation about the rest of the registers.

PSR

Processor Status Register is a read only register and contains the flags explained below. Bits 15 through 9 are reserved for future extensions.

-	ENW3	ENW2	ENW1	ENW0	IE	IL	RSWR	RSRD	UM
15...9	8	7	6	5	4	3	2	1	0

ENW3... ENW0: These bits enable the usage of cop interface wait signals. When set, the usage of the corresponding wait signal is enabled. When cleared, the wait signal is not used. See coprocessor interface document.

IE = 1: Interrupts enabled, IE = 0: Interrupts disabled.
IL = 1: Instruction word length is 32 bits, IL = 0: Instruction word length is 16 bits.
RSWR bit selects to which register set to write result:
RSWR = 1: SET2, privileged users set; RSWR = 0: SET1, users set.
RSRD bit selects from which register set to read operands:
RSRD = 1: SET2, privileged users set; RSRD = 0: SET1, users set.
UM indicates which mode the processor is in:
UM = 0: privileged user mode, UM = 1 : user mode.

SPSR

IE and IL flags are copied from SPSR to PSR when retu instruction is executed..
See 'Register usage of a privileged user' for information about how to use SPSR.

SPRs in SET 2

N	mnemonic	width	description/usage	notes
PR0	COP0_INT	32	Service routine starting address for coprocessor number 0 exceptions and interrupts	See 'coprocessor interface'
PR1	COP1_INT	32	Service routine starting address for coprocessor number 1 exceptions and interrupts	See 'coprocessor interface'
PR2	COP2_INT	32	Service routine starting address for coprocessor number 2 exceptions and interrupts	See 'coprocessor interface'
PR3	COP3_INT	32	Service routine starting address for coprocessor number 3 exceptions and interrupts	See 'coprocessor interface'
PR4	INTV0	32	Interrupt service routine 0 base address	See 'interrupts'
PR5	INTV1	32	Interrupt service routine 1 base address	See 'interrupts'
PR6	INTV2	32	Interrupt service routine 2 base address	See 'interrupts'
PR7	INTV3	32	Interrupt service routine 3 base address	See 'interrupts'
PR8	INTV4	32	Interrupt service routine 4 base address	See 'interrupts'
PR9	INTV5	32	Interrupt service routine 5 base address	See 'interrupts'
PR10	INTV6	32	Interrupt service routine 6 base address	See 'interrupts'

PR11	INTV7	32	Interrupt service routine 7 base address	See ‘interrupts’
PR12	INT_MOD	12	Interrupt service routine 0-7 mode flags and coprosessor interrupt mode flags	See ‘interrupts’
PR13	PSR	16	Prosessor status register	Bits 4…0 are read only
PR14	PSR_2	5	Copy of processor status register	Writable
PR15	ECS	4	Exception cause register	read only, see ‘exceptions’
PR16	EPC	32	Exception address	read only, see ‘exceptions’
PR17... PR30	PR17...PR30	32	General purpose registers	
PR31	PR31	32	General purpose register / link register	

CRs

There’s eight three bit wide condition registers C0...C7 visible both to application software and privileged software. Condition registers are used with conditional branches or when executing instructions conditionally. Each register contains three flags: Z (Zero), N (Negative) and C (Carry). When executing compare instructions or some arithmetic instructions these three flags are calculated and saved to the selected CR (arithmetic instructions allways save flags to C0). When conditionally branching or executing, flags from the selected CR are compared to match a certain condition given by the programmer. See chapters ‘conditonal execution’ and ‘instruction specifications’ for more infomation.

SFRs

The following registers are not directly user accessible. Information about these internal registers helps to get the most out of COFFEE. Internal registers not described here are irrelevant for the user or are explained in another more appropriate context.

PC – Program Counter

Contains the memory address of the next instruction to be decoded and executed. The address in PC is provided to instruction cache which fetches the instruction.

PEND

This register is internal to interrupt service logic (ISL). All interrupt requests which have gone through priority check but not yet served are pending in this register. See interrupts for more details about interrupts.

SERV

All interrupt sources being currently served (or interrupted by a source having greater priority) have a flag set in this register.

Register usage of a privileged user

When processor starts executing instructions from the address *start_addr_c* (see 'configuring core before synthesis') following conditions are assumed: 32 bit instruction word length, privileged user, register set SET 2 for reading and writing and interrupts disabled. Privileged user has the responsibility to initialize the special purpose registers to guarantee proper handling of interrupts and coprocessor exceptions. The privileged user passes control to application program by issuing the command *retu* (see 'instruction defininions' for details). Before passing the control, registers PSR2 and PR31 must be set appropiately. Executing *retu* causes PSR to be overwritten by PSR2 and PC overwritten by PR31. That is, execution will start at address saved to PR31 and with status flags saved in PSR2 before issuing the command *retu*.
When an application program issues the command *scall* (requesting some system service, for example), PSR2 is overwritten with PSR and PR31 is overwritten with link address (an address to return when resuming application code). In practise this means that privileged user is able to see the state in which the user was before calling system code and is able to resume execution from the correct address. Also the privileged user has full control over the user and the possibility to read and alter the status bits of the user.
An application program can pass parameters to privileged software (and the other way around) in some general purpose register RXX , if desired , since privileged software can read and write both sets of registers with the help of *chrs* command. For more information about instructions *scall*, *retu* and *chrs* see 'instruction defininions'.

Register limitations in 16 bit mode

In 16 bit instruction mode only the last eight registers from both sets are available, that is registers R24...R31 from set 1 and PR24...PR31 from set 2. Registers are mapped so that referring to register R0/PR0 in 16 bit mode means referring to register R24/PR24 in 32 bit mode and in general referring to Rx/PRx in 16 bit mode means referring to R(x+24)/PR(x+24) in 32 bit mode where x is an integer in the range 0...7.
Condition registers C1...C7 are disabled in 16 bit mode. Register C0 is always used (automatically selected) with conditional branches and arithmetic.

Register values after reset

PSR start value is 0000 0000 0000 1110b. Other registers are set to zero upon reset.

-	ENW3	ENW2	ENW1	ENW0	IE	IL	RSWR	RSRD	UM
15...9	8	7	6	5	4	3	2	1	0