Exceptions

priority	code	name	description	block & step
8	1000	instruction address violation *	While in user mode, instruction is beeing fetched from memory address not allowed for user.	INS_ADDR_CHECK, signalled in step 1
5	0101	unknown opcode	Unidentified command.	DECODE, signalled in step 2
6	0110	illegal opcode in user or 16bit mode	Trying to execute a command which is not valid in current operation mode.	DECODE, signalled in step 2
9	1001	miss aligned jump *	Calculated jump target is not aligned to word(32 bit mode) or halfword(16 bit mode) boundary.	INS_ADDR_CHECK, signalled in step 1
7	0111	trap	processor encountered a trap instruction	DECODE, signalled in step 2
3	0011	arithmetic overflow	The result a signed arithmetic operation exceeds 2^{31}-1.	ALU(ADD_SUB_UNIT, signalled in step 3
2	0010	arithmetic underflow	The result of a signed arithmetic operation falls below -2^{31}.	ALU(ADD_SUB_UNIT) signalled in step 3
0	0000	data address violation	While in user mode, a data address refers to memory address not allowed for user.	ALU(ADDR_CHECK), signalled in step 3
4	0100	jump address overflow	Trying to jump 'out' of the memory space below the bottom or above the top of the memory	RACU, signalled in step 2
1	0001	data address overflow	Trying to index data below of the bottom or above of the top of the memory	ALU(DATA_ADDR_CHK) , signalled in step 3

step 0 is fetch
step 1 is decode/operand fetch
step 2 execute one
step 3 execute two/memory access/CR access
step 4 execute three
step 5 write back

Priority 0 means most urgent and 9 means the lowest priority. Exceptions are served in the order of execution. When two or more instructions cause exceptions in different parts of the pipeline, the one with the highest priority is served. The offending instrucion and all following instrucions in the pipeline are flushed. If some instruction which entered the pipeline before the offending one, causes a new exception, the exception service routine will be restarted.
Offending instructions are not able to modify the state of the processor or contenst of the memory or registers.
There is one service routine common to all kinds of exceptions. The cause of the exception is saved in the special register ECS which is visible to the system code. * The memory address of the offending instruction is also saved in the EPC register, except for

the *instruction address violation* or *miss aligned jump* –exception, when the address referred to is saved. *Miss aligned jump* –exception can also be caused by improper interrupt address in one of the *int_vec* –registers. When entering the Exception service routine interrupts are disabled. They can be enabled again by the service routine, but this is not recommended. Also 32 bit super user mode is assumed and register sets for reading and writing are assumed to be super user sets.

When an exception occurs, interrupts are not served. The instruction, which caused the exception is not executed unless it changes the contents of a register or memory. All exceptions are served by the same routine, which starts at a specific address. The cause of the exception is saved at the register ECS. The state of the processor is saved in the register PSR_2 before the exception is served. The retu -instruction is used to return from the exception. If the user wants to continue at the next instruction following the instruction, which caused the exception, the address of the next instruction must be placed into the SR31 register. (The address can be calculated using status information found in PSR_2:IL and EPC registers.) The memory address of the instuction, which caused the exception is located in the EPC register, if the exception is any other than *instruction address violation*. In case of the *instruction address violation* the EPC register contains the current PC value. *Instruction address violation* may occur for example if the user tries to jump to the superuser address space or if the program execution goes to the superuser address space or if the exception vector points to the superuser address space, but the corresponding bit value in the MODE -register indicates that the exception was caused by user's instruction.