MIPSfpga+ External Interrupt Controller (IEC)

AUTHOR

Stanislav Zhelnio https://github.com/zhelnio

MAIN FEATURES

- very simple;
- up to 64 interrupts with independent interrupt vectors;
- the count of interrupts can be customized;
- two types of interrupt channels: direct channel and sense channel. The second type can be tuned to take only the low signal value, any logical change of input signal, the falling edge of input signal, or the rising edge of input signal.
- up to 32 sense channel interrupts;
- full support of microAptiv™ external interrupt controller operation options: 'Explicit Vector Number' and 'Explicit Vector Offset' - the interrupt handler offset can be directly transmitted to the CPU. For details see the chapter 5.3.1.3 in 'MIPS32® microAptiv™ UP Processor Core Family Software User's Manual, Revision 01.02';
- merged to MIPSfpga+ github project: **https://github.com/MIPSfpga/mipsfpga-plus**
- EIC usage example was included to mipsfpga-plus/**programs/07_iec**
- there is a standalone github project for controller debug: **https://github.com/zhelnio/ahb_lite_eic**
- to enable EIC uncomment option 'MFP_USE_IRQ_EIC' in **mfp_ahb_lite_matrix_config.vh** to set other setting use **mfp_eic_core.vh**
- to enable 'Explicit Vector Offset' option uncomment 'EIC_USE_EXPLICIT_VECTOR_OFFSET' in **mfp_eic_core.vh** and set 'assign eic_offset = 1'b1;' in **m14k_cpz_eicoffset_stub.v**

FILES

mfp_eic_core.v	IEC core;
mfp_eic_core.vh	main configuration file (see comments inside);
mfp_eic_handler.v	contains the logic to convert interrupt number to calling handler parameters: priority, vector number, offset;
mfp_eic_priority_encoder.v	tree structure of priority encoders;
mfp_ahb_lite_eic.v	top-level module, contains the interface to the system bus.

REGISTER DESCRIPTION

Name **EIC_REG_EICR**
Number 1
Description external interrupt control register
Operations read, write

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
reserved																															EICE

EICE enable external interrupt controller (asserts **SI_EICPresent** signal)
- 0 - external interrupt controller disabled
- 1 - external interrupt controller enabled

Name **EIC_REG_EIMSK_0, EIC_REG_EIMSK_1**
Number 2, 3
Description external interrupt mask register
Operations read, write

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IE31																															IE0

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
IE63																															IE32

IE**n** enable external interrupt **n**
- 0 - external interrupt disabled
- 1 - external interrupt enabled

Name **EIC_REG_EIFR_0, EIC_REG_EIFR_1**
Number 4, 5
Description external interrupt flag register
Operations read, write

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IF31																															IF0

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
IF63																															IF32

IF**n** interrupt request **n** is waiting for processing
- 0 - no interrupt request
- 1 - interrupt request is waiting for processing

Name **EIC_REG_EIFRS_0, EIC_REG_EIFRS_1**
Number 6, 7
Description external interrupt flag register, bit set
Operations write only

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IF31																															IF0

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
IF63																															IF32

IF**n** set interrupt flag **n**
0 - ignored
1 - set interrupt flag

Name **EIC_REG_EIFRC_0, EIC_REG_EIFRC_1**
Number 8, 9
Description external interrupt flag register, bit clear
Operations write only

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IF31																															IF0

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
IF63																															IF32

IF**n** clear interrupt flag **n**
0 - ignored
1 - clear interrupt flag

Name **EIC_REG_EISMSK_0, EIC_REG_EISMSK_1**
Number 10, 11
Description external interrupt sense mask register
Operations read, write

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SI16																														SI0	

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
SI31																														SI17	

SI**n** sense mask for channel **n** - only for sense channels
00 - the low level of input signal generates an interrupt request
01 - any logical change of input signal generates an interrupt request
10 - the falling edge of input signal generates an interrupt request
11 - the rising edge of input signal generates an interrupt request

---

Name **EIC_REG_EIIPR_0, EIC_REG_EIIPR_1**
Number 12, 13
Description external interrupt input pin register
Operations read only

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IP31																															IP0

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
IP63																															IP32

IP**n** interrupt **n** input pin state

Name **EIC_REG_EIACM_0, EIC_REG_EIACM_1**
Number 14, 15
Description external interrupt auto clear mask register
Operations read, write

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CF31																															CF0

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
CF63																															CF32

CF**n** clear interrupt flag **n**

0 - automatic clear function is disabled for interrupt **n**

1 - automatically clear interrupt **n** flag after signal **SI_IAck** receiving