

intMAN  
version 0.2\_20130422

Generated by Doxygen 1.8.3.1

Fri May 10 2013 12:06:53



# Contents

<b>1</b>	<b>Namespace Index</b>	<b>1</b>
<b>2</b>	<b>Design Unit Index</b>	<b>3</b>
2.1	Design Unit Hierarchy . . . . .	3
<b>3</b>	<b>Design Unit Index</b>	<b>5</b>
3.1	Design Unit List . . . . .	5
<b>4</b>	<b>File Index</b>	<b>7</b>
4.1	File List . . . . .	7
<b>5</b>	<b>Namespace Documentation</b>	<b>9</b>
5.1	intMAN_package Namespace Reference . . . . .	9
5.1.1	Detailed Description . . . . .	9
<b>6</b>	<b>Class Documentation</b>	<b>11</b>
6.1	intMAN_package Package Body Reference . . . . .	11
6.1.1	Detailed Description . . . . .	11
6.2	arch Architecture Reference . . . . .	11
6.2.1	Detailed Description . . . . .	12
6.2.2	Member Function Documentation . . . . .	12
6.2.2.1	p_icfg . . . . .	12
6.2.3	Member Data Documentation . . . . .	13
6.2.3.1	c_prio . . . . .	13
6.2.3.2	c_slack . . . . .	13
6.2.3.3	cpri . . . . .	13
6.2.3.4	cslack . . . . .	13
6.2.3.5	int_buf . . . . .	13
6.2.3.6	int_buf_in . . . . .	13
6.2.3.7	ip_high . . . . .	13
6.2.3.8	ipri_highest . . . . .	13
6.2.3.9	ipri_update_dmx . . . . .	14
6.2.3.10	ipupdmx . . . . .	14

6.2.3.11	SIG_ICFG	14
6.2.3.12	SIG_ICFG_MEM	14
6.2.3.13	SIG_ICNT	14
6.2.3.14	SIG_INTSEL	14
6.2.3.15	SIG_IPRIARR	14
6.2.3.16	SIG_IPRCOND_VALID	14
6.2.3.17	SIG_IPRIHVAL	14
6.2.3.18	SIG_IPRILOAD_SEL	15
6.2.3.19	SIG_IRDY	15
6.2.3.20	SIG_ISERV	15
6.2.3.21	SIG_SLACKCOND_VALID	15
6.3	arch Architecture Reference	15
6.3.1	Detailed Description	15
6.3.2	Member Function Documentation	15
6.3.2.1	p_cslack	15
6.4	arch Architecture Reference	16
6.4.1	Detailed Description	16
6.4.2	Member Function Documentation	16
6.4.2.1	PROCESS_6	16
6.5	arch Architecture Reference	16
6.5.1	Detailed Description	17
6.5.2	Member Function Documentation	17
6.5.2.1	newint_latch	17
6.5.2.2	PROCESS_1	18
6.5.3	Member Data Documentation	18
6.5.3.1	IEEE	18
6.5.3.2	intMAN_package	18
6.5.3.3	NUMERIC_STD	18
6.5.3.4	SIG_new_int	18
6.5.3.5	SIG_new_int_latch	18
6.5.3.6	signal_d	18
6.5.3.7	STD_LOGIC_1164	18
6.6	arch Architecture Reference	18
6.6.1	Detailed Description	19
6.6.2	Member Function Documentation	19
6.6.2.1	PROCESS_0	19
6.7	arch Architecture Reference	19
6.7.1	Detailed Description	20
6.7.2	Member Function Documentation	20
6.7.2.1	PROCESS_2	20

6.7.3	Member Data Documentation . . . . .	21
6.7.3.1	int_req . . . . .	21
6.7.3.2	intreq . . . . .	21
6.7.3.3	SIG_cnt . . . . .	21
6.7.3.4	SIG_int . . . . .	21
6.7.3.5	SIG_int_e . . . . .	21
6.7.3.6	SIG_int_l . . . . .	21
6.7.3.7	SIG_int_new . . . . .	21
6.7.3.8	SIG_pri . . . . .	22
6.7.3.9	SIG_rdy . . . . .	22
6.7.3.10	SIG_rst . . . . .	22
6.7.3.11	SIG_service . . . . .	22
6.7.3.12	signal_d . . . . .	22
6.8	arch Architecture Reference . . . . .	22
6.8.1	Detailed Description . . . . .	23
6.8.2	Member Function Documentation . . . . .	23
6.8.2.1	PROCESS_3 . . . . .	23
6.8.2.2	PROCESS_4 . . . . .	23
6.8.2.3	PROCESS_5 . . . . .	23
6.8.3	Member Data Documentation . . . . .	24
6.8.3.1	SIG_IPRIHIGH . . . . .	24
6.8.3.2	SIG_IPRIPOS . . . . .	24
6.8.3.3	SIG_IPRIPOSVALID . . . . .	24
6.9	c_prio Entity Reference . . . . .	24
6.9.1	Detailed Description . . . . .	25
6.9.2	Member Data Documentation . . . . .	25
6.9.2.1	hstipri . . . . .	25
6.9.2.2	intMAN_package . . . . .	25
6.9.2.3	monpri . . . . .	25
6.9.2.4	NUMERIC_STD . . . . .	25
6.9.2.5	valid . . . . .	25
6.10	c_slack Entity Reference . . . . .	26
6.10.1	Detailed Description . . . . .	26
6.10.2	Member Data Documentation . . . . .	26
6.10.2.1	intMAN_package . . . . .	26
6.10.2.2	monpri . . . . .	26
6.10.2.3	monslack . . . . .	27
6.10.2.4	NUMERIC_STD . . . . .	27
6.10.2.5	valid . . . . .	27
6.11	int_buf_in Entity Reference . . . . .	27

6.11.1 Detailed Description . . . . .	28
6.11.2 Member Data Documentation . . . . .	28
6.11.2.1 clk . . . . .	28
6.11.2.2 edge . . . . .	28
6.11.2.3 int_cnt . . . . .	29
6.11.2.4 int_in . . . . .	29
6.11.2.5 int_out . . . . .	29
6.11.2.6 int_pri . . . . .	29
6.11.2.7 int_pri_load . . . . .	29
6.11.2.8 int_pri_new . . . . .	29
6.11.2.9 int_rdy . . . . .	29
6.11.2.10 int_sel . . . . .	29
6.11.2.11 level . . . . .	29
6.11.2.12 reset . . . . .	30
6.12 int_req Entity Reference . . . . .	30
6.12.1 Detailed Description . . . . .	31
6.12.2 Member Data Documentation . . . . .	31
6.12.2.1 clk . . . . .	31
6.12.2.2 edge . . . . .	31
6.12.2.3 int_in . . . . .	31
6.12.2.4 int_new . . . . .	31
6.12.2.5 int_rdy . . . . .	31
6.12.2.6 intMAN_package . . . . .	31
6.12.2.7 level . . . . .	31
6.12.2.8 NUMERIC_STD . . . . .	32
6.12.2.9 reset . . . . .	32
6.13 intMAN_design Entity Reference . . . . .	32
6.13.1 Detailed Description . . . . .	33
6.13.2 Member Data Documentation . . . . .	33
6.13.2.1 CLK_1M . . . . .	33
6.13.2.2 INT_ACK . . . . .	33
6.13.2.3 INT_IN . . . . .	33
6.13.2.4 INT_OUT . . . . .	34
6.13.2.5 intMAN_package . . . . .	34
6.13.2.6 IPRI_LD . . . . .	34
6.13.2.7 IPRI_LD_ADDR . . . . .	34
6.13.2.8 IPRI_LD_DATA . . . . .	34
6.13.2.9 MON_CTX . . . . .	34
6.13.2.10 MON_INT . . . . .	34
6.13.2.11 MON_PRI . . . . .	34

6.13.2.12 MON_SLACK . . . . .	34
6.13.2.13 MON_TICK . . . . .	35
6.13.2.14 NUMERIC_STD . . . . .	35
6.13.2.15 RST . . . . .	35
6.14 intMAN_package Package Reference . . . . .	35
6.14.1 Detailed Description . . . . .	36
6.14.2 Member Data Documentation . . . . .	37
6.14.2.1 C_INTCFG . . . . .	37
6.14.2.2 CLK_BASE_HZ . . . . .	37
6.14.2.3 INT_C_MAX . . . . .	37
6.14.2.4 INT_C_MAX_NMONTICKS . . . . .	37
6.14.2.5 INT_C_MAX_NOSTICKS . . . . .	37
6.14.2.6 INT_NMAX . . . . .	37
6.14.2.7 INT_WIDTH . . . . .	37
6.14.2.8 MONCLK_HZ . . . . .	37
6.14.2.9 NMONTICKS_IN_OSTICK . . . . .	38
6.14.2.10 OSTIME_HZ . . . . .	38
6.14.2.11 PRI_HARD_LAST . . . . .	38
6.14.2.12 PRI_LAST . . . . .	38
6.14.2.13 PRI_WIDTH . . . . .	38
6.14.2.14 t_int_range . . . . .	38
6.14.2.15 t_int_width_range . . . . .	38
6.14.2.16 t_intcfg . . . . .	38
6.14.2.17 t_intcfg_arr . . . . .	38
6.14.2.18 t_intedge_cfg . . . . .	39
6.14.2.19 t_intlevel_cfg . . . . .	39
6.14.2.20 t_intpend_cnt . . . . .	39
6.14.2.21 t_intpendcnt_arr . . . . .	39
6.14.2.22 t_intpri_arr . . . . .	39
6.14.2.23 t_intrdy_vec . . . . .	39
6.14.2.24 t_pri_range . . . . .	39
6.14.2.25 t_pri_width_range . . . . .	39
6.15 ipri_highest Entity Reference . . . . .	39
6.15.1 Detailed Description . . . . .	40
6.15.2 Member Data Documentation . . . . .	40
6.15.2.1 intMAN_package . . . . .	40
6.15.2.2 IPRI_ADEC . . . . .	40
6.15.2.3 IPRI_ARR . . . . .	41
6.15.2.4 IPRI_HIGH . . . . .	41
6.15.2.5 NUMERIC_STD . . . . .	41

6.16	ipri_update_dmx Entity Reference	41
6.16.1	Detailed Description	42
6.16.2	Member Data Documentation	42
6.16.2.1	int_addr	42
6.16.2.2	intMAN_package	42
6.16.2.3	ipri_load_sel	42
6.16.2.4	NUMERIC_STD	42
6.16.2.5	sel	42
<b>7</b>	<b>File Documentation</b>	<b>43</b>
7.1	c_prio.vhd File Reference	43
7.1.1	Detailed Description	43
7.2	c_slack.vhd File Reference	43
7.2.1	Detailed Description	44
7.3	int_buf_in.vhd File Reference	44
7.3.1	Detailed Description	44
7.4	intman_design_top.vhd File Reference	44
7.4.1	Detailed Description	45
7.5	intman_pack.vhd File Reference	45
7.5.1	Detailed Description	45
7.6	ipri_highest.vhd File Reference	45
7.6.1	Detailed Description	46
7.7	ipri_update_dmx.vhd File Reference	46
7.7.1	Detailed Description	46
	<b>Index</b>	<b>46</b>



## **Chapter 1**

# **Namespace Index**



## Chapter 2

# Design Unit Index

### 2.1 Design Unit Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

intMAN_design . . . . .	32
ipri_update_dmx . . . . .	41
int_buf_in . . . . .	27
int_req . . . . .	30
ipri_highest . . . . .	39
c_prio . . . . .	24
c_slack . . . . .	26



## Chapter 3

# Design Unit Index

### 3.1 Design Unit List

Here is a list of all design unit members with links to the Entities they belong to:

architecture <a href="#">arch</a>	
IntMAN unit architecture (inner structure) . . . . .	11
architecture <a href="#">arch</a>	
Architecture of the slack-condition check unit for the intMAN hardware (inner structure) . . . .	15
architecture <a href="#">arch</a>	
Architecture of the interrupt address dmx unit for the intMAN hardware (inner structure) . . . .	16
architecture <a href="#">arch</a>	
Architecture of the unit for detection/latch of an interrupt request (inner structure) . . . . .	16
architecture <a href="#">arch</a>	
Architecture of the priority-condition check unit for the intMAN hardware (inner structure) . . . .	18
architecture <a href="#">arch</a>	
Architecture of the interrupt detect/buffer/config/forward unit for the intMAN hardware (inner structure) . . . . .	19
architecture <a href="#">arch</a>	
Architecture of the highest-priority pending interrupt select unit for the intMAN hardware (inner structure) . . . . .	22
entity <a href="#">c_prio</a>	
Entity of the priority-condition check unit for the intMAN hardware (interface) . . . . .	24
entity <a href="#">c_slack</a>	
Entity of the slack-condition check unit for the intMAN hardware (interface) . . . . .	26
entity <a href="#">int_buf_in</a>	
Entity of the interrupt detect/buffer/config/forward unit for the intMAN hardware (interface) . . . .	27
entity <a href="#">int_req</a>	
Entity of the unit for detection/latch of an interrupt request (interface) . . . . .	30
entity <a href="#">intMAN_design</a>	
IntMAN unit entity (interface) . . . . .	32
entity <a href="#">ipri_highest</a>	
Entity of the highest-priority pending interrupt select unit for the intMAN hardware (interface) . .	39
entity <a href="#">ipri_update_dmx</a>	
Entity of the interrupt address dmx unit for the intMAN hardware (interface) . . . . .	41



## Chapter 4

# File Index

### 4.1 File List

Here is a list of all files with brief descriptions:

<a href="#">c_prio.vhd</a>	Priority-condition check for the intMAN hardware . . . . .	43
<a href="#">c_slack.vhd</a>	Slack-condition check for the intMAN hardware . . . . .	43
<a href="#">int_buf_in.vhd</a>	Interrupt detect/buffer/config/forward for the intMAN hardware . . . . .	44
<a href="#">intman_design_top.vhd</a>	Top design of the intMAN hardware . . . . .	44
<a href="#">intman_pack.vhd</a>	Package of supplements for the intMAN hardware . . . . .	45
<a href="#">ipri_highest.vhd</a>	Highest-priority pending interrupt selector for the intMAN hardware . . . . .	45
<a href="#">ipri_update_dmx.vhd</a>	Interrupt address dmx unit for the intMAN hardware . . . . .	46





## Chapter 5

# Namespace Documentation

### 5.1 intMAN\_package Namespace Reference

custom definitions utilized during the intMAN implementation

#### 5.1.1 Detailed Description

custom definitions utilized during the intMAN implementation



## Chapter 6

# Class Documentation

### 6.1 intMAN\_package Package Body Reference

empty

#### 6.1.1 Detailed Description

empty

Definition at line 377 of file intman\_pack.vhd.

The documentation for this class was generated from the following file:

- [intman\\_pack.vhd](#)

### 6.2 arch Architecture Reference

intMAN unit architecture (inner structure)

#### Processes

- [p\\_icfg\( RST \)](#)  
*slack-condition result (signal)*

#### Components

- [int\\_buf\\_in](#)  
*interrupt detect/buffer/config/forward for the intMAN hardware*
- [ipri\\_update\\_dmx](#)  
*line to forward the interrupt*
- [ipri\\_highest](#)  
*address dmx output with no more than one interrupt selected*
- [c\\_prio](#)  
*address dmx output with the highest-priority interrupt selected*
- [c\\_slack](#)  
*priority-condition result*

## Signals

- **SIG\_IPRILOAD\_SEL** **STD\_LOGIC\_VECTOR**(**t\_int\_range** )  
*slack-condition result*
- **SIG\_ISERV** **STD\_LOGIC\_VECTOR**(**t\_int\_range** )  
*priority config update enable (signal)*
- **SIG\_ICNT** **t\_intpendcnt\_arr**  
*interrupt service start (signal)*
- **SIG\_IRDY** **t\_intrdy\_vec**  
*pending interrupt count (signal)*
- **SIG\_ICFG\_MEM** **t\_intcfg\_arr**  
*interrupt ready (signal)*
- **SIG\_ICFG** **t\_intcfg\_arr**  
*interrupt config storage (signal)*
- **SIG\_IPRIARR** **t\_intpri\_arr**  
*interrupt config bus (signal)*
- **SIG\_IPRIHVAL** **STD\_LOGIC\_VECTOR**(**t\_pri\_width\_range** )  
*interrupt priorities (signal)*
- **SIG\_INTSEL** **STD\_LOGIC\_VECTOR**(**t\_int\_range** )  
*the highest pending-interrupt priority (signal)*
- **SIG\_IPRICOND\_VALID** **STD\_LOGIC**  
*address dmx output with the highest-priority interrupt selected (signal)*
- **SIG\_SLACKCOND\_VALID** **STD\_LOGIC**  
*priority-condition result (signal)*

## Instantiations

- **ipupdmx** **ipri\_update\_dmx**  
*interrupt config storage unit init*
- **int\_buf** **int\_buf\_in**  
*interrupt detect/buffer/config/forward unit (port map)*
- **ip\_high** **ipri\_highest**  
*highest-priority pending interrupt unit (port map)*
- **cpri** **c\_prio**  
*priority condition check unit (port map)*
- **clack** **c\_slack**  
*priority condition check unit (port map)*

### 6.2.1 Detailed Description

intMAN unit architecture (inner structure)

Definition at line 41 of file intman\_design\_top.vhd.

### 6.2.2 Member Function Documentation

#### 6.2.2.1 **p\_icfg**( **RST** ) [Process]

slack-condition result (signal)

Definition at line 146 of file intman\_design\_top.vhd.

```

00146  p_icfg : process( RST ) is
00147  begin
00148  if( RST = ' 1 ' ) then
00149  SIG_ICFG_MEM  <=& C_INTCFG ;    -- after-reset init
00150  end if;
00151  end process p_icfg ;

```

## 6.2.3 Member Data Documentation

### 6.2.3.1 c\_prio [Component]

address dmx output with the highest-priority interrupt selected

Definition at line 83 of file intman\_design\_top.vhd.

### 6.2.3.2 c\_slack [Component]

priority-condition result

Definition at line 92 of file intman\_design\_top.vhd.

### 6.2.3.3 cpri c\_prio [Instantiation]

priority condition check unit (port map)

Definition at line 200 of file intman\_design\_top.vhd.

### 6.2.3.4 cslack c\_slack [Instantiation]

priority condition check unit (port map)

Definition at line 209 of file intman\_design\_top.vhd.

### 6.2.3.5 int\_buf int\_buf\_in [Instantiation]

interrupt detect/buffer/config/forward unit (port map)

Definition at line 172 of file intman\_design\_top.vhd.

### 6.2.3.6 int\_buf\_in [Component]

interrupt detect/buffer/config/forward for the intMAN hardware

Definition at line 44 of file intman\_design\_top.vhd.

### 6.2.3.7 ip\_high ipri\_highest [Instantiation]

highest-priority pending interrupt unit (port map)

Definition at line 191 of file intman\_design\_top.vhd.

### 6.2.3.8 ipri\_highest [Component]

address dmx output with no more than one interrupt selected

Definition at line 73 of file intman\_design\_top.vhd.

#### 6.2.3.9 **ipri\_update\_dmx** [Component]

line to forward the interrupt

Definition at line 63 of file intman\_design\_top.vhd.

#### 6.2.3.10 **ipupdmx ipri\_update\_dmx** [Instantiation]

interrupt config storage unit init

interrupt config selector unit (port map)

Definition at line 162 of file intman\_design\_top.vhd.

#### 6.2.3.11 **SIG\_ICFG t\_intcfg\_arr** [Signal]

interrupt config storage (signal)

Definition at line 122 of file intman\_design\_top.vhd.

#### 6.2.3.12 **SIG\_ICFG\_MEM t\_intcfg\_arr** [Signal]

interrupt ready (signal)

Definition at line 119 of file intman\_design\_top.vhd.

#### 6.2.3.13 **SIG\_ICNT t\_intpendcnt\_arr** [Signal]

interrupt service start (signal)

Definition at line 113 of file intman\_design\_top.vhd.

#### 6.2.3.14 **SIG\_INTSEL STD.LOGIC.VECTOR(t\_int\_range)** [Signal]

the highest pending-interrupt priority (signal)

Definition at line 131 of file intman\_design\_top.vhd.

#### 6.2.3.15 **SIG\_IPRIARR t\_intpri\_arr** [Signal]

interrupt config bus (signal)

Definition at line 125 of file intman\_design\_top.vhd.

#### 6.2.3.16 **SIG\_IPRICOND\_VALID STD.LOGIC** [Signal]

address dmx output with the highest-priority interrupt selected (signal)

Definition at line 134 of file intman\_design\_top.vhd.

#### 6.2.3.17 **SIG\_IPRIHVAL STD.LOGIC.VECTOR(t\_pri\_width\_range)** [Signal]

interrupt priorities (signal)

Definition at line 128 of file intman\_design\_top.vhd.

**6.2.3.18 SIG\_IPRILOAD\_SEL STD\_LOGIC\_VECTOR(t\_int\_range) [Signal]**

slack-condition result

Definition at line 107 of file intman\_design\_top.vhd.

**6.2.3.19 SIG\_IRDY t\_intrdy\_vec [Signal]**

pending interrupt count (signal)

Definition at line 116 of file intman\_design\_top.vhd.

**6.2.3.20 SIG\_ISERV STD\_LOGIC\_VECTOR(t\_int\_range) [Signal]**

priority config update enable (signal)

Definition at line 110 of file intman\_design\_top.vhd.

**6.2.3.21 SIG\_SLACKCOND\_VALID STD\_LOGIC [Signal]**

priority-condition result (signal)

Definition at line 137 of file intman\_design\_top.vhd.

The documentation for this class was generated from the following file:

- [intman\\_design\\_top.vhd](#)

## 6.3 arch Architecture Reference

Architecture of the slack-condition check unit for the intMAN hardware (inner structure)

### Processes

- [p\\_cslack\( monpri , monslack \)](#)

*Slack-condition check process being sensitive to the change of monitored priority and slack line.*

### 6.3.1 Detailed Description

Architecture of the slack-condition check unit for the intMAN hardware (inner structure)

Definition at line 33 of file c\_slack.vhd.

### 6.3.2 Member Function Documentation

**6.3.2.1 p\_cslack( monpri , monslack ) [Process]**

Slack-condition check process being sensitive to the change of monitored priority and slack line.

Definition at line 36 of file c\_slack.vhd.

```
00036  p_cslack : process( monpri , monslack )
00037  begin
00038  if ((TO_INTEGER(UNSIGNED( monpri )) >= PRI_HARD_LAST) or ( monslack = ' 1 ')) then
00039    valid <= ' 1 ';
00040  else
```

```

00041   valid  <=> ' 0 ' ;
00042   end if;
00043   end process;

```

The documentation for this class was generated from the following file:

- [c\\_slack.vhd](#)

## 6.4 arch Architecture Reference

Architecture of the interrupt address dmx unit for the intMAN hardware (inner structure)

### Processes

- [PROCESS\\_6\( sel , int\\_addr \)](#)  
*selection signal generation process*

#### 6.4.1 Detailed Description

Architecture of the interrupt address dmx unit for the intMAN hardware (inner structure)

Definition at line 33 of file ipri\_update\_dmx.vhd.

#### 6.4.2 Member Function Documentation

##### 6.4.2.1 [PROCESS\\_6\( sel , int\\_addr \)](#) [Process]

selection signal generation process

Definition at line 36 of file ipri\_update\_dmx.vhd.

```

00036   process( sel , int_addr ) is
00037   variable idx : integer;
00038   variable j : STD_LOGIC_VECTOR( t_int_range );
00039   begin
00040   idx := TO_INTEGER(UNSIGNED( int_addr ));
00041   j := (OTHERS => ' 0 ');
00042   if ( sel = ' 1 ' ) then
00043   if(idx <=> INT_NMAX - 1 ) then
00044   j(idx) := ' 1 ' ;
00045   end if;
00046   end if;
00047   ipri_load_sel <=> j;
00048   end process;

```

-- put ' 1 ' into the position determined by int-

The documentation for this class was generated from the following file:

- [ipri\\_update\\_dmx.vhd](#)

## 6.5 arch Architecture Reference

Architecture of the unit for detection/latch of an interrupt request (inner structure)



## Processes

- `PROCESS_1( clk )`  
*process for detection/capturing of an arriving interrupt*
- `newint_latch( reset , SIG_new_int )`  
*process for latching the pending interrupt request*

## Libraries

- `IEEE`

## Use Clauses

- `STD_LOGIC_1164`
- `NUMERIC_STD`
- `intMAN_package`

## Signals

- `signal_d STD_LOGIC`  
*D-storage.*
- `SIG_new_int STD_LOGIC`  
*new int line*
- `SIG_new_int_latch STD_LOGIC`  
*new int flag/latch*

### 6.5.1 Detailed Description

Architecture of the unit for detection/latch of an interrupt request (inner structure)

Definition at line 36 of file `int_buf_in.vhd`.

### 6.5.2 Member Function Documentation

#### 6.5.2.1 `newint_latch( reset , SIG_new_int )` [Process]

process for latching the pending interrupt request

Definition at line 66 of file `int_buf_in.vhd`.

```
00066  newint_latch : process ( reset , SIG_new_int )
00067  begin
00068    if ( SIG_new_int = ' 1 ' ) then
00069      SIG_new_int_latch <= ' 1 ' ;
00070    elsif ( reset = ' 1 ' ) then
00071      SIG_new_int_latch <= ' 0 ' ;
00072    end if;
00073  end process;
```

### 6.5.2.2 PROCESS\_1( clk ) [Process]

process for detection/capturing of an arriving interrupt

Definition at line 51 of file int\_buf\_in.vhd.

```
00051 process( clk )
00052 begin
00053   if ( clk 'event and clk = ' 1 ' ) then           -- rising edge
00054     signal_d <=<= int_in ;
00055   end if;
00056 end process;
```

## 6.5.3 Member Data Documentation

### 6.5.3.1 IEEE [Library]

Definition at line 82 of file int\_buf\_in.vhd.

### 6.5.3.2 intMAN\_package [Package]

Definition at line 85 of file int\_buf\_in.vhd.

### 6.5.3.3 NUMERIC\_STD [Package]

Definition at line 84 of file int\_buf\_in.vhd.

### 6.5.3.4 SIG\_new\_int STD.LOGIC [Signal]

new int line

Definition at line 40 of file int\_buf\_in.vhd.

### 6.5.3.5 SIG\_new\_int\_latch STD.LOGIC [Signal]

new int flag/latch

Definition at line 42 of file int\_buf\_in.vhd.

### 6.5.3.6 signal\_d STD.LOGIC [Signal]

D-storage.

Definition at line 38 of file int\_buf\_in.vhd.

### 6.5.3.7 STD\_LOGIC\_1164 [Package]

Definition at line 83 of file int\_buf\_in.vhd.

The documentation for this class was generated from the following file:

- [int\\_buf\\_in.vhd](#)

## 6.6 arch Architecture Reference

Architecture of the priority-condition check unit for the intMAN hardware (inner structure)

## Processes

- [PROCESS\\_0\( monpri , hstipri \)](#)

*Priority-condition check process being sensitive to the change of pending interrupt and monitored task priorities.*

### 6.6.1 Detailed Description

Architecture of the priority-condition check unit for the intMAN hardware (inner structure)

Definition at line 32 of file c\_prio.vhd.

### 6.6.2 Member Function Documentation

#### 6.6.2.1 PROCESS\_0( monpri , hstipri ) [Process]

Priority-condition check process being sensitive to the change of pending interrupt and monitored task priorities.

Definition at line 35 of file c\_prio.vhd.

```
00035 process( monpri , hstipri )
00036 begin
00037   if( hstipri <= monpri ) then
00038     valid <= ' 1 ' ;
00039   else
00040     valid <= ' 0 ' ;
00041   end if;
00042 end process;
```

The documentation for this class was generated from the following file:

- [c\\_prio.vhd](#)

## 6.7 arch Architecture Reference

Architecture of the interrupt detect/buffer/config/forward unit for the intMAN hardware (inner structure)

## Processes

- [PROCESS\\_2\( reset , clk , int\\_pri\\_load , int\\_sel \)](#)  
*control process of the interrupt detect/buffer/config/forward unit*

## Components

- [int\\_req](#)  
*component for detection/latch of an interrupt request*

## Signals

- [signal\\_d](#) **STD\_LOGIC**  
*pending/ready interrupt flag*
- [SIG\\_rdy](#) **STD\_LOGIC**  
*D-latch.*
- [SIG\\_rst](#) **STD\_LOGIC**  
*interrupt ready*

- **SIG\_cnt t\_intpend\_cnt**  
*interrupt request reset*
- **SIG\_pri STD\_LOGIC\_VECTOR(t\_pri\_width\_range)**  
*pending interrupts counter*
- **SIG\_int\_l STD\_LOGIC**  
*priority*
- **SIG\_int\_e STD\_LOGIC**  
*level match flag*
- **SIG\_int STD\_LOGIC**  
*edge match flag*
- **SIG\_int\_new STD\_LOGIC**  
*incomming interrupt line*
- **SIG\_service STD\_LOGIC**  
*new interrupt flag*

## Instantiations

- **intreq int\_req**  
*to service forward flag*

## 6.7.1 Detailed Description

Architecture of the interrupt detect/buffer/config/forward unit for the intMAN hardware (inner structure)

Definition at line 114 of file int\_buf\_in.vhd.

## 6.7.2 Member Function Documentation

### 6.7.2.1 PROCESS\_2( reset, clk, int\_pri\_load, int\_sel ) [get]

control process of the interrupt detect/buffer/config/forward unit

Definition at line 174 of file int\_buf\_in.vhd.

```

174
175     process(reset, clk, int_pri_load, int_sel)
176     begin
177         if(reset = '1') then                                -- reset (async.)
178             SIG_cnt <= 0;
179         elsif(SIG_int_new'event and SIG_int_new = '1') then -- new interrupt arrival (async.)
180             SIG_cnt <= SIG_cnt+1;
181         elsif(clk'event and clk='1') then                   -- clk-synchronous storage
182             operations
183                 if (int_pri_load = '1') then                -- ... interrupt priority
184                     update enable
185                         SIG_pri <= int_pri_new;
186                     elsif(int_sel = '1') then                -- ... if an interrupt
187                         is to be forwarded, signalize it
188                             SIG_service <= '1';
189                         if (SIG_cnt > 0) then
190                             SIG_cnt <= SIG_cnt - 1;          -- ... and
191                         decrease the pending-interrupt counter
192                             end if;
193                         else
194                             SIG_service <= '0';
195                         end if;
196                     -- interrupt sensitivity legend:
197                     -----
198                     -- level: log.0 (00), log. 1 (01), no (10, 11)
199                     -- edge: rising (01), falling (10), both (11), no (00)
200
201                     case level is

```

```

200             when "00" => SIG_int_l <= ('0' or not int_sel);
201             when "01" => SIG_int_l <= ('1' and int_sel);
202             when others => SIG_int_l <= SIG_int_l or '0';
203         end case;
204
205         case edge is
206             when "01" => SIG_int_e <= ('1' and int_sel);
207             when "10" => SIG_int_e <= ('0' or not int_sel);
208             when "11" => SIG_int_e <= ('1' and int_sel);
209             when others => SIG_int_e <= SIG_int_e or '0';
210         end case;
211
212         int_out <= SIG_int_l or SIG_int_e;           -- forward the interrupt
213     according to its config
214     end if;
215 end process;

```

### 6.7.3 Member Data Documentation

#### 6.7.3.1 **int\_req** [Component]

component for detection/latch of an interrupt request

Definition at line 117 of file int\_buf\_in.vhd.

#### 6.7.3.2 **intreq int\_req** [Instantiation]

to service forward flag

mapping ports to the interrupt request detect/latch unit (port map)

Definition at line 159 of file int\_buf\_in.vhd.

#### 6.7.3.3 **SIG\_cnt t.intpend.cnt** [Signal]

interrupt request reset

Definition at line 136 of file int\_buf\_in.vhd.

#### 6.7.3.4 **SIG\_int STD.LOGIC** [Signal]

edge match flag

Definition at line 145 of file int\_buf\_in.vhd.

#### 6.7.3.5 **SIG\_int\_e STD.LOGIC** [Signal]

level match flag

Definition at line 143 of file int\_buf\_in.vhd.

#### 6.7.3.6 **SIG\_int\_l STD.LOGIC** [Signal]

priority

Definition at line 141 of file int\_buf\_in.vhd.

#### 6.7.3.7 **SIG\_int\_new STD.LOGIC** [Signal]

incomming interrupt line

Definition at line 147 of file int\_buf\_in.vhd.

### 6.7.3.8 SIG\_pri **STD\_LOGIC\_VECTOR**(t\_pri\_width\_range) [Signal]

pending interrupts counter

Definition at line 138 of file int\_buf\_in.vhd.

### 6.7.3.9 SIG\_rdy **STD\_LOGIC** [Signal]

D-latch.

Definition at line 132 of file int\_buf\_in.vhd.

### 6.7.3.10 SIG\_rst **STD\_LOGIC** [Signal]

interrupt ready

Definition at line 134 of file int\_buf\_in.vhd.

### 6.7.3.11 SIG\_service **STD\_LOGIC** [Signal]

new interrupt flag

Definition at line 149 of file int\_buf\_in.vhd.

### 6.7.3.12 signal\_d **STD\_LOGIC** [Signal]

pending/ready interrupt flag

Definition at line 130 of file int\_buf\_in.vhd.

The documentation for this class was generated from the following files:

- [int\\_buf\\_in.vhd](#)

## 6.8 arch Architecture Reference

Architecture of the highest-priority pending interrupt select unit for the intMAN hardware (inner structure)

### Processes

- [PROCESS\\_3\( IPRI\\_ARR \)](#)  
*Process of searching the highest priority of pending interrupts.*
- [PROCESS\\_4\( IPRI\\_ARR \)](#)  
*store the HPRI interrupt index*
- [PROCESS\\_5\( IPRI\\_ARR \)](#)  
*Process of SIG\_IPRIPOSVALID adjustment.*

### Signals

- [SIG\\_IPRIHIGH \*\*STD\\_LOGIC\\_VECTOR\*\*\(t\\_pri\\_width\\_range \)](#)  
*highest priority (HPRI) value signal*
- [SIG\\_IPRIPOS \*\*STD\\_LOGIC\\_VECTOR\*\*\(t\\_int\\_width\\_range \)](#)  
*HPRI interrupt index signal.*

- **SIG\_IPRIPOSVALID STD\_LOGIC**

*HPRI interrupt index validity signal.*

### 6.8.1 Detailed Description

Architecture of the highest-priority pending interrupt select unit for the intMAN hardware (inner structure)

Definition at line 32 of file ipri\_highest.vhd.

### 6.8.2 Member Function Documentation

#### 6.8.2.1 PROCESS.3( IPRI\_ARR ) [Process]

Process of searching the highest priority of pending interrupts.

Definition at line 41 of file ipri\_highest.vhd.

```

00041 process( IPRI_ARR ) is
00042   variable highest_pri : STD_LOGIC_VECTOR( t_pri_width_range );    --! highest priority (HPR-
I) value
00043   variable highest_pos : STD_LOGIC_VECTOR( t_int_width_range );    --! HPRI interrupt index
00044   begin
00045     highest_pri := (OTHERS =>$ ' 1 ');                                --! init to the lowest (' 1 ... 1 ') priority value
00046     highest_pos := (OTHERS =>$ ' 0 ');                                --! reset index
00047     for i in t_int_range loop
00048       if( IPRI_ARR (i) < highest_pri ) then                          --! new HPRI found
00049         highest_pri := IPRI_ARR (i);                                --! store the HPRI value
00050         highest_pos := STD_LOGIC_VECTOR( TO_UNSIGNED(i, INT_WIDTH )); --! store the HPRI interrupt index
00051       end if;
00052     end loop;
00053
00054     SIG_IPRIHIGH  <=& highest_pri;
00055     SIG_IPRIPOS   <=& highest_pos;
00056   end process;

```

#### 6.8.2.2 PROCESS.4( IPRI\_ARR ) [Process]

store the HPRI interrupt index

Definition at line 60 of file ipri\_highest.vhd.

```

00060 process( IPRI_ARR ) is
00061   variable i: integer;
00062   variable j: STD_LOGIC;
00063   begin
00064     j := ' 0 ';
00065     for i in t_pri_width_range loop
00066       j := (j or (not SIG_IPRIHIGH (i)));
00067     end loop;
00068     SIG_IPRIPOSVALID <=& j;
00069   end process;

```

PRI field

-- index is valid if there is at least one ' 0 '

#### 6.8.2.3 PROCESS.5( IPRI\_ARR ) [Process]

Process of SIG\_IPRIPOSVALID adjustment.

Definition at line 74 of file ipri\_highest.vhd.

```

00074 process( IPRI_ARR ) is
00075   variable i: integer;
00076   begin
00077     i := TO_INTEGER(UNSIGNED( SIG_IPRIPOS ));
00078
00079     FOR idx IN t_int_range LOOP
00080       if(idx = i) then

```

```

00081  IPRI_ADEC (idx) <= ( ' 1 ' and SIG_IPRIPOSVALID );           --! make select-line active only if the H-
PRI index is valid
00082  else
00083  IPRI_ADEC (idx) <= ' 0 ' ;
00084  end if;
00085  end LOOP;
00086  end process;

```

### 6.8.3 Member Data Documentation

#### 6.8.3.1 SIG\_IPRIHIGH **STD\_LOGIC\_VECTOR**(t\_pri\_width\_range) [Signal]

highest priority (HPRI) value signal

Definition at line 34 of file ipri\_highest.vhd.

#### 6.8.3.2 SIG\_IPRIPOS **STD\_LOGIC\_VECTOR**(t\_int\_width\_range) [Signal]

HPRI interrupt index signal.

Definition at line 36 of file ipri\_highest.vhd.

#### 6.8.3.3 SIG\_IPRIPOSVALID **STD\_LOGIC** [Signal]

HPRI interrupt index validity signal.

Definition at line 38 of file ipri\_highest.vhd.

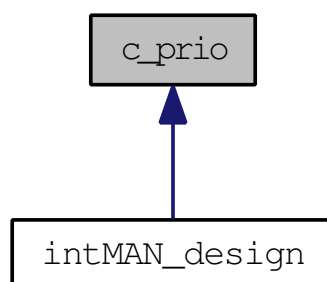
The documentation for this class was generated from the following file:

- [ipri\\_highest.vhd](#)

## 6.9 c\_prio Entity Reference

Entity of the priority-condition check unit for the intMAN hardware (interface)

Inheritance diagram for c\_prio:



### Entities

- [arch](#) architecture

*Architecture of the priority-condition check unit for the intMAN hardware (inner structure)*



## Use Clauses

- [NUMERIC\\_STD](#)
- [intMAN\\_package](#)

## Ports

- **monpri** in **STD\_LOGIC\_VECTOR**(**t\_pri\_width\_range**)  
*priority of the running task (being monitored)*
- **hstipri** in **STD\_LOGIC\_VECTOR**(**t\_pri\_width\_range**)  
*the highest pending-interrupt priority*
- **valid** out **STD\_LOGIC**  
*priority-condition result*

### 6.9.1 Detailed Description

Entity of the priority-condition check unit for the intMAN hardware (interface)

Definition at line 19 of file c\_prio.vhd.

### 6.9.2 Member Data Documentation

#### 6.9.2.1 **hstipri** in **STD\_LOGIC\_VECTOR**(**t\_pri\_width\_range**) [Port]

the highest pending-interrupt priority

Definition at line 22 of file c\_prio.vhd.

#### 6.9.2.2 **intMAN\_package** [Package]

Definition at line 12 of file c\_prio.vhd.

#### 6.9.2.3 **monpri** in **STD\_LOGIC\_VECTOR**(**t\_pri\_width\_range**) [Port]

priority of the running task (being monitored)

Definition at line 21 of file c\_prio.vhd.

#### 6.9.2.4 **NUMERIC\_STD** [Package]

Definition at line 11 of file c\_prio.vhd.

#### 6.9.2.5 **valid** out **STD\_LOGIC** [Port]

priority-condition result

Definition at line 23 of file c\_prio.vhd.

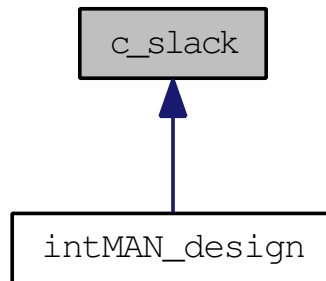
The documentation for this class was generated from the following file:

- [c\\_prio.vhd](#)

## 6.10 c\_slack Entity Reference

Entity of the slack-condition check unit for the intMAN hardware (interface)

Inheritance diagram for c\_slack:



### Entities

- [arch](#) architecture  
*Architecture of the slack-condition check unit for the intMAN hardware (inner structure)*

### Use Clauses

- [NUMERIC\\_STD](#)
- [intMAN\\_package](#)

### Ports

- [monpri](#) **in** [STD\\_LOGIC\\_VECTOR](#)([t\\_pri\\_width\\_range](#))  
*priority of the running task (being monitored)*
- [monslack](#) **in** [STD\\_LOGIC](#)  
*slack signal (being monitored)*
- [valid](#) **out** [STD\\_LOGIC](#)  
*priority-condition result*

### 6.10.1 Detailed Description

Entity of the slack-condition check unit for the intMAN hardware (interface)

Definition at line 20 of file c\_slack.vhd.

### 6.10.2 Member Data Documentation

#### 6.10.2.1 intMAN\_package [Package]

Definition at line 13 of file c\_slack.vhd.

#### 6.10.2.2 monpri **in** [STD\\_LOGIC\\_VECTOR](#)([t\\_pri\\_width\\_range](#)) [Port]

priority of the running task (being monitored)

Definition at line 22 of file c\_slack.vhd.

### 6.10.2.3 monslack in STD\_LOGIC [Port]

slack signal (being monitored)

Definition at line 23 of file c\_slack.vhd.

### 6.10.2.4 NUMERIC\_STD [Package]

Definition at line 11 of file c\_slack.vhd.

### 6.10.2.5 valid out STD\_LOGIC [Port]

priority-condition result

Definition at line 24 of file c\_slack.vhd.

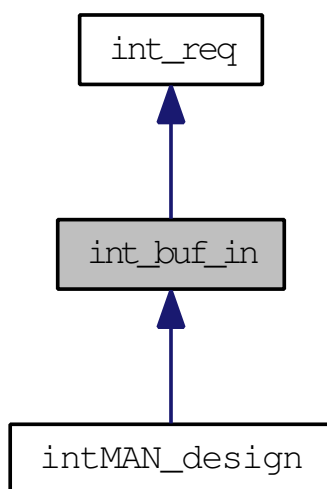
The documentation for this class was generated from the following file:

- [c\\_slack.vhd](#)

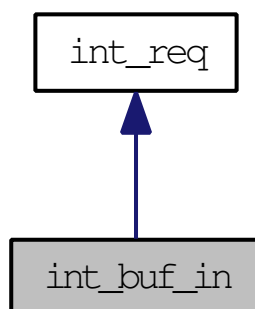
## 6.11 int\_buf\_in Entity Reference

Entity of the interrupt detect/buffer/config/forward unit for the intMAN hardware (interface)

Inheritance diagram for int\_buf\_in:



Collaboration diagram for int\_buf\_in:



## Entities

- [arch](#) architecture

*Architecture of the interrupt detect/buffer/config/forward unit for the intMAN hardware (inner structure)*

## Ports

- [reset](#) **in** **STD\_LOGIC**  
*async reset*
- [level](#) **in** [t\\_intlevel\\_cfg](#)  
*level-sensitivity select: log.0 (00), log. 1 (01), no (10, 11)*
- [edge](#) **in** [t\\_intedge\\_cfg](#)  
*edge-sensitivity select: rising (01), falling (10), both (11), no (00)*
- [clk](#) **in** **STD\_LOGIC**  
*clock*
- [int\\_in](#) **in** **STD\_LOGIC**  
*incomming interrupt line*
- [int\\_rdy](#) **out** **STD\_LOGIC**  
*line to signalize there is a ready/pending interrupt*
- [int\\_cnt](#) **out** [t\\_intpend\\_cnt](#)  
*number of ready/pending interrupts*
- [int\\_pri\\_load](#) **in** **STD\_LOGIC**  
*priority update enable/disable*
- [int\\_pri\\_new](#) **in** **STD\_LOGIC\_VECTOR**([t\\_pri\\_width\\_range](#))  
*new interrupt priority*
- [int\\_pri](#) **out** **STD\_LOGIC\_VECTOR**([t\\_pri\\_width\\_range](#))  
*interrupt priority storage*
- [int\\_sel](#) **in** **STD\_LOGIC**  
*interrupt forward logic enable*
- [int\\_out](#) **out** **STD\_LOGIC**  
*line to forward the interrupt*

### 6.11.1 Detailed Description

Entity of the interrupt detect/buffer/config/forward unit for the intMAN hardware (interface)

Definition at line 92 of file `int_buf_in.vhd`.

### 6.11.2 Member Data Documentation

#### 6.11.2.1 [clk](#) **in** **STD\_LOGIC** [Port]

clock

Definition at line 97 of file `int_buf_in.vhd`.

#### 6.11.2.2 [edge](#) **in** [t\\_intedge\\_cfg](#) [Port]

edge-sensitivity select: rising (01), falling (10), both (11), no (00)

Definition at line 96 of file `int_buf_in.vhd`.

**6.11.2.3 int\_cnt out t.intpend\_cnt [Port]**

number of ready/pending interrupts

Definition at line 100 of file int\_buf\_in.vhd.

**6.11.2.4 int\_in in STD\_LOGIC [Port]**

incomming interrupt line

Definition at line 98 of file int\_buf\_in.vhd.

**6.11.2.5 int\_out out STD\_LOGIC [Port]**

line to forward the interrupt

Definition at line 105 of file int\_buf\_in.vhd.

**6.11.2.6 int\_pri out STD\_LOGIC\_VECTOR(t.pri.width\_range) [Port]**

interrupt priority storage

Definition at line 103 of file int\_buf\_in.vhd.

**6.11.2.7 int\_pri\_load in STD\_LOGIC [Port]**

priority update enable/disable

Definition at line 101 of file int\_buf\_in.vhd.

**6.11.2.8 int\_pri\_new in STD\_LOGIC\_VECTOR(t.pri.width\_range) [Port]**

new interrupt priority

Definition at line 102 of file int\_buf\_in.vhd.

**6.11.2.9 int\_rdy out STD\_LOGIC [Port]**

line to signalize there is a ready/pending interrupt

Definition at line 99 of file int\_buf\_in.vhd.

**6.11.2.10 int\_sel in STD\_LOGIC [Port]**

interrupt forward logic enable

Definition at line 104 of file int\_buf\_in.vhd.

**6.11.2.11 level in t.intlevel.cfg [Port]**

level-sensitivity select: log.0 (00), log. 1 (01), no (10, 11)

Definition at line 95 of file int\_buf\_in.vhd.

#### 6.11.2.12 `reset` in `STD_LOGIC` [Port]

async reset

Definition at line 94 of file `int_buf_in.vhd`.

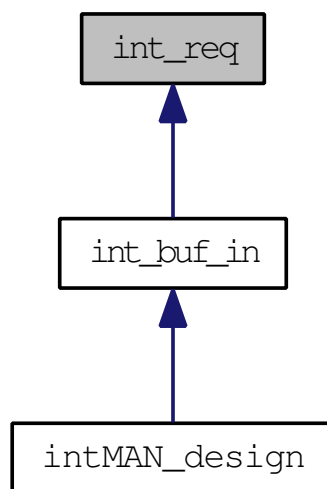
The documentation for this class was generated from the following file:

- [int\\_buf\\_in.vhd](#)

## 6.12 `int_req` Entity Reference

Entity of the unit for detection/latch of an interrupt request (interface)

Inheritance diagram for `int_req`:



### Entities

- [arch](#) architecture

*Architecture of the unit for detection/latch of an interrupt request (inner structure)*

### Use Clauses

- [NUMERIC\\_STD](#)
- [intMAN\\_package](#)

### Ports

- `reset` in `STD_LOGIC`  
*async. reset*
- `level` in `t_intlevel_cfg`  
*level-sensitivity select: log.0 (00), log. 1 (01), no (10, 11)*
- `edge` in `t_intedge_cfg`  
*edge-sensitivity select: rising (01), falling (10), both (11), no (00)*
- `clk` in `STD_LOGIC`  
*clock*

- **int\_in in STD\_LOGIC**  
*incomming interrupt line*
- **int\_new out STD\_LOGIC**  
*new interrupt-arrival flag*
- **int\_rdy out STD\_LOGIC**  
*pending/ready interrupt flag*

### 6.12.1 Detailed Description

Entity of the unit for detection/latch of an interrupt request (interface)

Definition at line 19 of file int\_buf\_in.vhd.

### 6.12.2 Member Data Documentation

#### 6.12.2.1 **clk in STD\_LOGIC** [Port]

clock

Definition at line 24 of file int\_buf\_in.vhd.

#### 6.12.2.2 **edge in t\_intedge\_cfg** [Port]

edge-sensitivity select: rising (01), falling (10), both (11), no (00)

Definition at line 23 of file int\_buf\_in.vhd.

#### 6.12.2.3 **int\_in in STD\_LOGIC** [Port]

incomming interrupt line

Definition at line 25 of file int\_buf\_in.vhd.

#### 6.12.2.4 **int\_new out STD\_LOGIC** [Port]

new interrupt-arrival flag

Definition at line 26 of file int\_buf\_in.vhd.

#### 6.12.2.5 **int\_rdy out STD\_LOGIC** [Port]

pending/ready interrupt flag

Definition at line 27 of file int\_buf\_in.vhd.

#### 6.12.2.6 **intMAN\_package** [Package]

Definition at line 12 of file int\_buf\_in.vhd.

#### 6.12.2.7 **level in t\_intlevel\_cfg** [Port]

level-sensitivity select: log.0 (00), log. 1 (01), no (10, 11)

Definition at line 22 of file int\_buf\_in.vhd.

### 6.12.2.8 NUMERIC\_STD [Package]

Definition at line 11 of file int\_buf\_in.vhd.

### 6.12.2.9 reset in STD\_LOGIC [Port]

async. reset

Definition at line 21 of file int\_buf\_in.vhd.

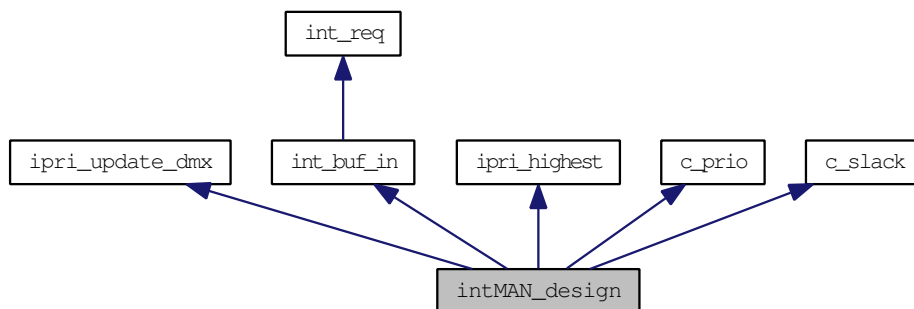
The documentation for this class was generated from the following file:

- [int\\_buf\\_in.vhd](#)

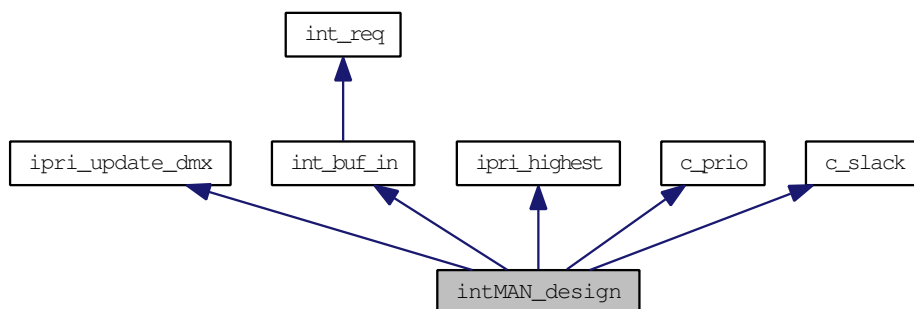
## 6.13 intMAN\_design Entity Reference

intMAN unit entity (interface)

Inheritance diagram for intMAN\_design:



Collaboration diagram for intMAN\_design:



## Entities

- [arch](#) architecture  
*intMAN unit architecture (inner structure)*

## Use Clauses

- [NUMERIC\\_STD](#)
- [intMAN\\_package](#)



## Ports

- **MON\_INT in STD\_LOGIC**  
*line to monitor whether the RT system is operating at the ISR (interrupt service routine) level*
- **MON\_TICK in STD\_LOGIC**  
*line to monitor whether the RT system is servicing the OS (operating system) time tick*
- **MON\_CTX in STD\_LOGIC**  
*line to monitor whether the RT system is performing a task-level context-switch*
- **MON\_PRI in STD\_LOGIC\_VECTOR(t\_pri\_width\_range)**  
*line to monitor the priority of a task running in the RT system*
- **MON\_SLACK in STD\_LOGIC**  
*line to monitor whether there is slack time in the RT schedule*
- **INT\_ACK in STD\_LOGIC**  
*line to acknowledge an INT service*
- **INT\_OUT out STD\_LOGIC\_VECTOR(t\_int\_range)**  
*line to forward pending interrupt requests to the RT system*
- **INT\_IN in STD\_LOGIC\_VECTOR(t\_int\_range)**  
*line to detect interrupt requests willing to stimulate the RT system*
- **IPRI\_LD in STD\_LOGIC**  
*line to enable update of an interrupt configuration*
- **IPRI\_LD\_ADDR in STD\_LOGIC\_VECTOR(t\_int\_width\_range)**  
*line to address the interrupt configuration of which is going to be updated*
- **IPRI\_LD\_DATA in STD\_LOGIC\_VECTOR(t\_pri\_width\_range)**  
*line to adjust data going to be written into the configuration*
- **CLK\_1M in STD\_LOGIC**  
*clock signal line*
- **RST in STD\_LOGIC**  
*reset signal line*

### 6.13.1 Detailed Description

intMAN unit entity (interface)

Definition at line 19 of file intman\_design\_top.vhd.

### 6.13.2 Member Data Documentation

#### 6.13.2.1 CLK\_1M in STD\_LOGIC [Port]

clock signal line

Definition at line 32 of file intman\_design\_top.vhd.

#### 6.13.2.2 INT\_ACK in STD\_LOGIC [Port]

line to acknowledge an INT service

Definition at line 26 of file intman\_design\_top.vhd.

#### 6.13.2.3 INT\_IN in STD\_LOGIC\_VECTOR(t\_int\_range) [Port]

line to detect interrupt requests willing to stimulate the RT system

Definition at line 28 of file intman\_design\_top.vhd.

#### 6.13.2.4 INT\_OUT out STD\_LOGIC\_VECTOR(t\_int\_range) [Port]

line to forward pending interrupt requests to the RT system

Definition at line 27 of file intman\_design\_top.vhd.

#### 6.13.2.5 intMAN\_package [Package]

Definition at line 12 of file intman\_design\_top.vhd.

#### 6.13.2.6 IPRI\_LD in STD\_LOGIC [Port]

line to enable update of an interrupt configuration

Definition at line 29 of file intman\_design\_top.vhd.

#### 6.13.2.7 IPRI\_LD\_ADDR in STD\_LOGIC\_VECTOR(t\_int\_width\_range) [Port]

line to address the interrupt configuration of which is going to be updated

Definition at line 30 of file intman\_design\_top.vhd.

#### 6.13.2.8 IPRI\_LD\_DATA in STD\_LOGIC\_VECTOR(t\_pri\_width\_range) [Port]

line to adjust data going to be written into the configuration

Definition at line 31 of file intman\_design\_top.vhd.

#### 6.13.2.9 MON\_CTX in STD\_LOGIC [Port]

line to monitor whether the RT system is performing a task-level context-switch

Definition at line 23 of file intman\_design\_top.vhd.

#### 6.13.2.10 MON\_INT in STD\_LOGIC [Port]

line to monitor whether the RT system is operating at the ISR (interrupt service routine) level

Definition at line 21 of file intman\_design\_top.vhd.

#### 6.13.2.11 MON\_PRI in STD\_LOGIC\_VECTOR(t\_pri\_width\_range) [Port]

line to monitor the priority of a task running in the RT system

Definition at line 24 of file intman\_design\_top.vhd.

#### 6.13.2.12 MON\_SLACK in STD\_LOGIC [Port]

line to monitor whether there is slack time in the RT schedule

Definition at line 25 of file intman\_design\_top.vhd.

## 6.13.2.13 MON\_TICK in STD\_LOGIC [Port]

line to monitor whether the RT system is servicing the OS (operating system) time tick

Definition at line 22 of file intman\_design\_top.vhd.

## 6.13.2.14 NUMERIC\_STD [Package]

Definition at line 11 of file intman\_design\_top.vhd.

## 6.13.2.15 RST in STD\_LOGIC [Port]

reset signal line

Definition at line 33 of file intman\_design\_top.vhd.

The documentation for this class was generated from the following file:

- [intman\\_design\\_top.vhd](#)

## 6.14 intMAN\_package Package Reference

custom definitions utilized during the intMAN implementation

**Package Body** >> [intMAN\\_package](#)

### Constants

- **CLK\_BASE\_HZ integer := 10 \*\* 6**  
*base clock rate [Hz]*
- **MONCLK\_HZ integer := 1 \* CLK\_BASE\_HZ**  
*monitor clock (MONCLK) rate [Hz]*
- **OSTIME\_HZ integer := 100**  
*operating system time (OSTIME) rate [Hz]*
- **NMONTICKS\_IN\_OSTICK integer := MONCLK\_HZ / OSTIME\_HZ**  
*number of MONCLK cycles (ticks) per one OSTIME cycle (tick)*
- **PRI\_WIDTH integer := 8**  
*number of bits per joint interrupt/task priority*
- **PRI\_HARD\_LAST integer := ( 2 \*\* PRI\_WIDTH ) /**  
*value at which the hard-level priority ends (0 means the highest priority)*
- **PRI\_LAST integer := ( 2 \*\* PRI\_WIDTH ) -**  
*the lowest priority (reserved for the intMAN purposes and set to the all-1 value)*
- **INT\_C\_MAX real := 150 . 0 / REAL( CLK\_BASE\_HZ )**  
*the worst-case interrupt service time [s]*
- **INT\_C\_MAX\_NMONTICKS integer := INTEGER( INT\_C\_MAX ) \* MONCLK\_HZ**  
*the worst-case interrupt service time [MONCLK cycles]*
- **INT\_C\_MAX\_NOSTICKS integer := INT\_C\_MAX\_NMONTICKS / NMONTICKS\_IN\_OSTICK**  
*the worst-case interrupt service time [OSTIME cycles]*
- **INT\_NMAX integer := 16**  
*number of interrupt sources*
- **INT\_WIDTH integer := 4**

*number of bits needed to address the interrupt sources ( $=\log_2(\text{INT\_NMAX})$ )*

- **C\_INTCFG t\_intcfg\_arr** :=( 0 =>(level=>" 00 ",edge=>" 11 "), 1 =>(level=>" 00 ",edge=>" 11 "), 2 =>(level=>" 00 ",edge=>" 11 "), 3 =>(level=>" 00 ",edge=>" 11 "), 4 =>(level=>" 00 ",edge=>" 00 "), 5 =>(level=>" 00 ",edge=>" 00 "), 6 =>(level=>" 00 ",edge=>" 11 "), 7 =>(level=>" 00 ",edge=>" 10 "), 8 =>(level=>" 00 ",edge=>" 01 "), 9 =>(level=>" 00 ",edge=>" 01 "), 10 =>(level=>" 00 ",edge=>" 01 "), 11 =>(level=>" 00 ",edge=>" 01 "), 12 =>(level=>" 00 ",edge=>" 01 "), 13 =>(level=>" 00 ",edge=>" 01 "), 14 =>(level=>" 00 ",edge=>" 01 "), 15 =>(level=>" 00 ",edge=>" 01 "))

*interrupt configuration*

## Types

- **t\_intpendcnt\_arr**array(t\_int\_range)oft\_intpend\_cnt  
*array of pending interrupt couters*
- **record: t\_intcfg**level:t\_intlevel\_cfg  
edge:t\_intedge\_cfg  
*interrupt sensitivity configuration data structure*
- **t\_intcfg\_arr**array(t\_int\_range)oft\_intcfg  
*array of configurations for interrupt sources*
- **t\_intpri\_arr**array(t\_int\_range)ofSTD\_LOGIC\_VECTOR(t\_pri\_width\_range )  
*array of interrupt priorities*

## Subtypes

- **t\_pri\_width\_range** integerrangePRI\_WIDTH - 1 downto 0  
*bit-index range for the priority width*
- **t\_pri\_range** integerrange( 2 \*\*PRI\_WIDTH )- 1 downto 0  
*priority value range*
- **t\_int\_width\_range** integerrangeINT\_WIDTH - 1 downto 0  
*interrupt address bit-index range*
- **t\_int\_range** integerrangeINT\_NMAX - 1 downto 0  
*interrupt range*
- **t\_intpend\_cnt** integerrange 0 to 15  
*maximum number of pending interrupts per one interrupt source*
- **t\_intrdy\_vec** STD\_LOGIC\_VECTOR(t\_int\_range )  
*vector of interrupt ready flags*
- **t\_intlevel\_cfg** STD\_LOGIC\_VECTOR( 1 downto 0 )  
*level sensitivity configuration datatype*
- **t\_intedge\_cfg** STD\_LOGIC\_VECTOR( 1 downto 0 )  
*edge sensitivity configuration datatype*

### 6.14.1 Detailed Description

custom definitions utilized during the intMAN implementation

Definition at line 17 of file intman\_pack.vhd.

## 6.14.2 Member Data Documentation

**6.14.2.1 C\_INTCFG** `t_intcfg_arr := ( 0 =>(level=>"00",edge=>"11"), 1 =>(level=>"00",edge=>"11"), 2 =>(level=>"00",edge=>"11"), 3 =>(level=>"00",edge=>"11"), 4 =>(level=>"00",edge=>"00"), 5 =>(level=>"00",edge=>"00"), 6 =>(level=>"00",edge=>"11"), 7 =>(level=>"00",edge=>"10"), 8 =>(level=>"00",edge=>"01"), 9 =>(level=>"00",edge=>"01"), 10 =>(level=>"00",edge=>"01"), 11 =>(level=>"00",edge=>"01"), 12 =>(level=>"00",edge=>"01"), 13 =>(level=>"00",edge=>"01"), 14 =>(level=>"00",edge=>"01"), 15 =>(level=>"00",edge=>"01"))` [Constant]

interrupt configuration

Definition at line 98 of file intman\_pack.vhd.

**6.14.2.2 CLK\_BASE\_HZ** `integer:= 10 ** 6` [Constant]

base clock rate [Hz]

Definition at line 26 of file intman\_pack.vhd.

**6.14.2.3 INT\_C\_MAX** `real:= 150 . 0 /REAL(CLK_BASE_HZ)` [Constant]

the worst-case interrupt service time [s]

Definition at line 55 of file intman\_pack.vhd.

**6.14.2.4 INT\_C\_MAX\_NMONTICKS** `integer:=INTEGER(INT_C_MAX)*MONCLK_HZ` [Constant]

the worst-case interrupt service time [MONCLK cycles]

Definition at line 57 of file intman\_pack.vhd.

**6.14.2.5 INT\_C\_MAX\_NOSTICKS** `integer:=INT_C_MAX_NMONTICKS /NMONTICKS_IN_OSTICK` [Constant]

the worst-case interrupt service time [OSTIME cycles]

Definition at line 59 of file intman\_pack.vhd.

**6.14.2.6 INT\_NMAX** `integer:= 16` [Constant]

number of interrupt sources

Definition at line 62 of file intman\_pack.vhd.

**6.14.2.7 INT\_WIDTH** `integer:= 4` [Constant]

number of bits needed to address the interrupt sources (=log2(INT\_NMAX))

Definition at line 64 of file intman\_pack.vhd.

**6.14.2.8 MONCLK\_HZ** `integer:= 1 *CLK_BASE_HZ` [Constant]

monitor clock (MONCLK) rate [Hz]

Definition at line 28 of file intman\_pack.vhd.

#### 6.14.2.9 NMONTICKS\_IN\_OSTICK **integer:=MONCLK\_HZ /OSTIME\_HZ** [Constant]

number of MONCLK cycles (ticks) per one OSTIME cycle (tick)

Definition at line 32 of file intman\_pack.vhd.

#### 6.14.2.10 OSTIME\_HZ **integer:= 100** [Constant]

operating system time (OSTIME) rate [Hz]

Definition at line 30 of file intman\_pack.vhd.

#### 6.14.2.11 PRI\_HARD\_LAST **integer:=( 2 \*\*PRI\_WIDTH )/** [Constant]

value at which the hard-level priority ends (0 means the highest priority)

Definition at line 47 of file intman\_pack.vhd.

#### 6.14.2.12 PRI\_LAST **integer:=( 2 \*\*PRI\_WIDTH )-** [Constant]

the lowest priority (reserved for the intMAN purposes and set to the all-1 value)

Definition at line 49 of file intman\_pack.vhd.

#### 6.14.2.13 PRI\_WIDTH **integer:= 8** [Constant]

number of bits per joint interrupt/task priority

Definition at line 41 of file intman\_pack.vhd.

#### 6.14.2.14 t\_int\_range **integerrangeINT\_NMAX - 1 downto 0** [Subtype]

interrupt range

Definition at line 68 of file intman\_pack.vhd.

#### 6.14.2.15 t\_int\_width\_range **integerrangeINT\_WIDTH - 1 downto 0** [Subtype]

interrupt address bit-index range

Definition at line 66 of file intman\_pack.vhd.

#### 6.14.2.16 t\_intcfg (

interrupt sensitivity configuration data structure

Definition at line 83 of file intman\_pack.vhd.

#### 6.14.2.17 t\_intcfg\_arr **isarray(t\_int\_range )oft\_intcfg** [Type]

array of configurations for interrupt sources

Definition at line 90 of file intman\_pack.vhd.

6.14.2.18 `t_intedge_cfg` **STD\_LOGIC\_VECTOR**( `1 downto 0` ) [Subtype]

edge sensitivity configuration datatype

Definition at line 80 of file `intman_pack.vhd`.

6.14.2.19 `t_intlevel_cfg` **STD\_LOGIC\_VECTOR**( `1 downto 0` ) [Subtype]

level sensitivity configuration datatype

Definition at line 78 of file `intman_pack.vhd`.

6.14.2.20 `t_intpend_cnt` **integerrange** `0 to 15` [Subtype]

maximum number of pending interrupts per one interrupt source

Definition at line 71 of file `intman_pack.vhd`.

6.14.2.21 `t_intpendcnt_arr` **isarray**(`t_int_range`) **of** `t_intpend_cnt` [Type]

array of pending interrupt counters

Definition at line 73 of file `intman_pack.vhd`.

6.14.2.22 `t_intpri_arr` **isarray**(`t_int_range`) **of** **STD\_LOGIC\_VECTOR**(`t_pri_width_range`) [Type]

array of interrupt priorities

Definition at line 372 of file `intman_pack.vhd`.

6.14.2.23 `t_intrdy_vec` **STD\_LOGIC\_VECTOR**(`t_int_range`) [Subtype]

vector of interrupt ready flags

Definition at line 75 of file `intman_pack.vhd`.

6.14.2.24 `t_pri_range` **integerrange**( `2**PRI_WIDTH - 1 downto` ) [Subtype]

priority value range

Definition at line 45 of file `intman_pack.vhd`.

6.14.2.25 `t_pri_width_range` **integerrange** `PRI_WIDTH - 1 downto 0` [Subtype]

bit-index range for the priority width

Definition at line 43 of file `intman_pack.vhd`.

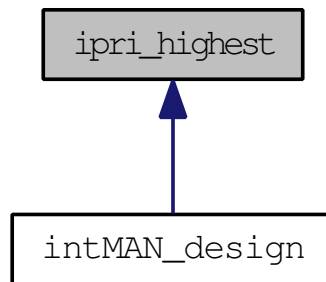
The documentation for this class was generated from the following file:

- [intman\\_pack.vhd](#)

## 6.15 ipri\_highest Entity Reference

Entity of the highest-priority pending interrupt select unit for the intMAN hardware (interface)

Inheritance diagram for ipri\_highest:



## Entities

- [arch](#) architecture

*Architecture of the highest-priority pending interrupt select unit for the intMAN hardware (inner structure)*

## Use Clauses

- [NUMERIC\\_STD](#)
- [intMAN\\_package](#)

## Ports

- [IPRI\\_ARR](#) in [t\\_intpri\\_arr](#)  
*interrupt priorities*
- [IPRI\\_HIGH](#) out [STD\\_LOGIC\\_VECTOR\(t\\_pri\\_width\\_range\)](#)  
*the highest interrupt priority*
- [IPRI\\_ADEC](#) out [STD\\_LOGIC\\_VECTOR\(t\\_int\\_range\)](#)  
*address dmx output with the highest-priority interrupt selected*

### 6.15.1 Detailed Description

Entity of the highest-priority pending interrupt select unit for the intMAN hardware (interface)

Definition at line 19 of file ipri\_highest.vhd.

### 6.15.2 Member Data Documentation

#### 6.15.2.1 intMAN\_package [Package]

Definition at line 12 of file ipri\_highest.vhd.

#### 6.15.2.2 IPRI\_ADEC out STD\_LOGIC\_VECTOR(t\_int\_range) [Port]

address dmx output with the highest-priority interrupt selected

Definition at line 23 of file ipri\_highest.vhd.



## 6.15.2.3 IPRI\_ARR in t\_intpri\_arr [Port]

interrupt priorities

Definition at line 21 of file ipri\_highest.vhd.

## 6.15.2.4 IPRI\_HIGH out STD\_LOGIC\_VECTOR(t\_pri\_width\_range) [Port]

the highest interrupt priority

Definition at line 22 of file ipri\_highest.vhd.

## 6.15.2.5 NUMERIC\_STD [Package]

Definition at line 11 of file ipri\_highest.vhd.

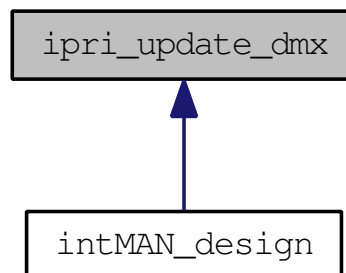
The documentation for this class was generated from the following file:

- [ipri\\_highest.vhd](#)

## 6.16 ipri\_update\_dmx Entity Reference

Entity of the interrupt address dmx unit for the intMAN hardware (interface)

Inheritance diagram for ipri\_update\_dmx:



### Entities

- [arch](#) architecture  
*Architecture of the interrupt address dmx unit for the intMAN hardware (inner structure)*

### Use Clauses

- [NUMERIC\\_STD](#)
- [intMAN\\_package](#)

### Ports

- [sel](#) in [std\\_logic](#)  
*unit enable*
- [int\\_addr](#) in [STD\\_LOGIC\\_VECTOR\(t\\_int\\_width\\_range\)](#)  
*selected interrupt address*

- `ipri_load_sel out STD_LOGIC_VECTOR(t_int_range)`

*address dmx output with no more than one interrupt selected*

### 6.16.1 Detailed Description

Entity of the interrupt address dmx unit for the intMAN hardware (interface)

Definition at line 20 of file `ipri_update.dmx.vhd`.

### 6.16.2 Member Data Documentation

#### 6.16.2.1 `int_addr in STD_LOGIC_VECTOR(t_int_width_range)` [Port]

selected interrupt address

Definition at line 23 of file `ipri_update.dmx.vhd`.

#### 6.16.2.2 `intMAN_package` [Package]

Definition at line 13 of file `ipri_update.dmx.vhd`.

#### 6.16.2.3 `ipri_load_sel out STD_LOGIC_VECTOR(t_int_range)` [Port]

address dmx output with no more than one interrupt selected

Definition at line 24 of file `ipri_update.dmx.vhd`.

#### 6.16.2.4 `NUMERIC_STD` [Package]

Definition at line 12 of file `ipri_update.dmx.vhd`.

#### 6.16.2.5 `sel in std_logic` [Port]

unit enable

Definition at line 22 of file `ipri_update.dmx.vhd`.

The documentation for this class was generated from the following file:

- `ipri_update.dmx.vhd`

# Chapter 7

## File Documentation

### 7.1 c\_prio.vhd File Reference

Priority-condition check for the intMAN hardware.

#### Entities

- [c\\_prio](#) entity  
*Entity of the priority-condition check unit for the intMAN hardware (interface)*
- [arch](#) architecture  
*Architecture of the priority-condition check unit for the intMAN hardware (inner structure)*

#### 7.1.1 Detailed Description

Priority-condition check for the intMAN hardware.

#### Author

Josef Strnadel, Brno University of Technology, Faculty of Information Technology [strnadel@fit.vutbr.cz](mailto:strnadel@fit.vutbr.cz)

#### Date

2013-04-17

Definition in file [c\\_prio.vhd](#).

### 7.2 c\_slack.vhd File Reference

Slack-condition check for the intMAN hardware.

#### Entities

- [c\\_slack](#) entity  
*Entity of the slack-condition check unit for the intMAN hardware (interface)*
- [arch](#) architecture  
*Architecture of the slack-condition check unit for the intMAN hardware (inner structure)*

### 7.2.1 Detailed Description

Slack-condition check for the intMAN hardware.

Author

Josef Strnadel, Brno University of Technology, Faculty of Information Technology [strnadel@fit.vutbr.cz](mailto:strnadel@fit.vutbr.cz)

Date

2013-04-17

Definition in file [c\\_slack.vhd](#).

## 7.3 int\_buf\_in.vhd File Reference

Interrupt detect/buffer/config/forward for the intMAN hardware.

### Entities

- [int\\_req](#) entity  
*Entity of the unit for detection/latch of an interrupt request (interface)*
- [arch](#) architecture  
*Architecture of the unit for detection/latch of an interrupt request (inner structure)*
- [int\\_buf\\_in](#) entity  
*Entity of the interrupt detect/buffer/config/forward unit for the intMAN hardware (interface)*
- [arch](#) architecture  
*Architecture of the interrupt detect/buffer/config/forward unit for the intMAN hardware (inner structure)*

### 7.3.1 Detailed Description

Interrupt detect/buffer/config/forward for the intMAN hardware.

Author

Josef Strnadel, Brno University of Technology, Faculty of Information Technology [strnadel@fit.vutbr.cz](mailto:strnadel@fit.vutbr.cz)

Date

2013-04-22

Definition in file [int\\_buf\\_in.vhd](#).

## 7.4 intman\_design\_top.vhd File Reference

Top design of the intMAN hardware.

### Entities

- [intMAN\\_design](#) entity  
*intMAN unit entity (interface)*
- [arch](#) architecture  
*intMAN unit architecture (inner structure)*

### 7.4.1 Detailed Description

Top design of the intMAN hardware.

#### Author

Josef Strnadel, Brno University of Technology, Faculty of Information Technology [strnadel@fit.vutbr.cz](mailto:strnadel@fit.vutbr.cz)

#### Date

2013-04-17

Definition in file [intman\\_design\\_top.vhd](#).

## 7.5 intman\_pack.vhd File Reference

Package of supplements for the intMAN hardware.

### Entities

- [intMAN\\_package](#) package  
*custom definitions utilized during the intMAN implementation*
- [intMAN\\_package](#) package body  
*empty*

### 7.5.1 Detailed Description

Package of supplements for the intMAN hardware.

#### Author

Josef Strnadel, Brno University of Technology, Faculty of Information Technology [strnadel@fit.vutbr.cz](mailto:strnadel@fit.vutbr.cz)

#### Date

2013-04-17

Definition in file [intman\\_pack.vhd](#).

## 7.6 ipri\_highest.vhd File Reference

Highest-priority pending interrupt selector for the intMAN hardware.

### Entities

- [ipri\\_highest](#) entity  
*Entity of the highest-priority pending interrupt select unit for the intMAN hardware (interface)*
- [arch](#) architecture  
*Architecture of the highest-priority pending interrupt select unit for the intMAN hardware (inner structure)*

### 7.6.1 Detailed Description

Highest-priority pending interrupt selector for the intMAN hardware.

#### Author

Josef Strnadel, Brno University of Technology, Faculty of Information Technology [strnadel@fit.vutbr.cz](mailto:strnadel@fit.vutbr.cz)

#### Date

2013-04-17

Definition in file [ipri\\_highest.vhd](#).

## 7.7 ipri\_update\_dmx.vhd File Reference

Interrupt address dmx unit for the intMAN hardware.

### Entities

- [ipri\\_update\\_dmx](#) entity  
*Entity of the interrupt address dmx unit for the intMAN hardware (interface)*
- [arch](#) architecture  
*Architecture of the interrupt address dmx unit for the intMAN hardware (inner structure)*

### 7.7.1 Detailed Description

Interrupt address dmx unit for the intMAN hardware. After it is enabled it activates the interrupt buffer identified by `int_addr` in order to perform further operation over it such as config update.

#### Author

Josef Strnadel, Brno University of Technology, Faculty of Information Technology [strnadel@fit.vutbr.cz](mailto:strnadel@fit.vutbr.cz)

#### Date

2013-04-17

Definition in file [ipri\\_update\\_dmx.vhd](#).