

XilPM Library v3.5



Table of Contents

Chapter 1: XilPM Zynq UltraScale+ MPSoC APIs.....	3
Functions.....	7
Enumerations.....	38
Definitions.....	43
Error Status.....	48
Chapter 2: XilPM Versal ACAP APIs.....	65
Functions.....	69
Enumerations.....	97
Definitions.....	108
Power Nodes.....	126
Reset Nodes.....	131
Clock Nodes.....	152
MIO Nodes.....	178
Device Nodes.....	196
Subsystem Nodes.....	216
Event Node IDs.....	216
Error Event Mask.....	218
Chapter 3: Library Parameters in MSS File.....	242
Chapter 4: Data Structure Index.....	243
pm_acknowledge.....	243
pm_init_suspend.....	244
XPm_DeviceStatus.....	244
XPm_NodeStatus.....	245
XPm_Notifier.....	245
Appendix A: Additional Resources and Legal Notices.....	247
Xilinx Resources.....	247
Documentation Navigator and Design Hubs.....	247
Please Read: Important Legal Notices.....	248

XilPM Zynq UltraScale+ MPSoC APIs

Table 1: Quick Function Reference

Type	Name	Arguments
XStatus	XPm_InitXilpm	XIpiPsu * IpiInst
enum	XPm_GetBootStatus	void
void	XPm_SuspendFinalize	void
XStatus	XPm_SelfSuspend	const enum XPmNodeId d nid const u32 latency const u8 state const u64 address
XStatus	XPm_SetConfiguration	const u32 address
XStatus	XPm_InitFinalize	void
XStatus	XPm_RequestSuspend	const enum XPmNodeId d target const enum XPmRequestAck ack const u32 latency const u8 state
XStatus	XPm_RequestWakeUp	const enum XPmNodeId d target const bool setAddress const u64 address const enum XPmRequestAck ack
XStatus	XPm_ForcePowerDown	const enum XPmNodeId d target const enum XPmRequestAck ack

Table 1: Quick Function Reference (cont'd)

Type	Name	Arguments
XStatus	XPm_AbortSuspend	const enum XPmAbortReason reason
XStatus	XPm_SetWakeUpSource	const enum XPmNodeId target const enum XPmNodeId wkup_node const u8 enable
XStatus	XPm_SystemShutdown	u32 type u32 subtype
XStatus	XPm_RequestNode	const enum XPmNodeId node const u32 capabilities const u32 qos const enum XPmRequestAck ack
XStatus	XPm_SetRequirement	const enum XPmNodeId nid const u32 capabilities const u32 qos const enum XPmRequestAck ack
XStatus	XPm_ReleaseNode	const enum XPmNodeId node
XStatus	XPm_SetMaxLatency	const enum XPmNodeId node const u32 latency
void	XPm_InitSuspendCb	const enum XPmSuspendReason reason const u32 latency const u32 state const u32 timeout
void	XPm_AcknowledgeCb	const enum XPmNodeId node const XStatus status const u32 oppoint
void	XPm_NotifyCb	const enum XPmNodeId node const enum XPmNotifyEvent event const u32 oppoint
XStatus	XPm_GetApiVersion	u32 * version
XStatus	XPm_GetNodeStatus	const enum XPmNodeId node XPm_NodeStatus *const nodestatus

Table 1: Quick Function Reference (cont'd)

Type	Name	Arguments
XStatus	XPm_GetOpCharacteristic	const enum XPmNodeId node const enum XPmOpCharType type u32 *const result
XStatus	XPm_ResetAssert	const enum XPmReset reset const enum XPmResetAction resetaction
XStatus	XPm_ResetGetStatus	const enum XPmReset reset u32 * status
XStatus	XPm_RegisterNotifier	XPmNotifier *const notifier
XStatus	XPm_UnregisterNotifier	XPmNotifier *const notifier
XStatus	XPm_MmioWrite	const u32 address const u32 mask const u32 value
XStatus	XPm_MmioRead	const u32 address u32 *const value
XStatus	XPm_ClockEnable	const enum XPmClock clk
XStatus	XPm_ClockDisable	const enum XPmClock clk
XStatus	XPm_ClockGetStatus	const enum XPmClock clk u32 *const status
XStatus	XPm_ClockSetOneDivider	const enum XPmClock clk const u32 divider const u32 divId
XStatus	XPm_ClockSetDivider	const enum XPmClock clk const u32 divider
XStatus	XPm_ClockGetOneDivider	const enum XPmClock clk u32 *const divider const u32 divId

Table 1: Quick Function Reference (cont'd)

Type	Name	Arguments
XStatus	XPm_ClockGetDivider	const enum XPmCl ock clk u32 *const divider
XStatus	XPm_ClockSetParent	const enum XPmCl ock clk const enum XPmCl ock parent
XStatus	XPm_ClockGetParent	const enum XPmCl ock clk enum XPmCl ock *const parent
XStatus	XPm_ClockSetRate	const enum XPmCl ock clk const u32 rate
XStatus	XPm_ClockGetRate	const enum XPmCl ock clk u32 *const rate
XStatus	XPm_PllSetParameter	const enum XPmNodeI d node const enum XPmPI l Par am parameter const u32 value
XStatus	XPm_PllGetParameter	const enum XPmNodeI d node const enum XPmPI l Par am parameter u32 *const value
XStatus	XPm_PllSetMode	const enum XPmNodeI d node const enum XPmPI l M ode mode
XStatus	XPm_PllGetMode	const enum XPmNodeI d node enum XPmPI l M ode *const mode
XStatus	XPm_PinCtrlAction	const u32 pin const enum XPmApi l d api
XStatus	XPm_PinCtrlRequest	const u32 pin
XStatus	XPm_PinCtrlRelease	const u32 pin
XStatus	XPm_PinCtrlSetFunction	const u32 pin const enum XPmPi nFn fn

Table 1: Quick Function Reference (cont'd)

Type	Name	Arguments
XStatus	XPm_PinCtrlGetFunction	const u32 pin enum XPnP_i_nFn *const fn
XStatus	XPm_PinCtrlSetParameter	const u32 pin const enum XPnP_i_nPar amparam const u32 value
XStatus	XPm_PinCtrlGetParameter	const u32 pin const enum XPnP_i_nPar amparam u32 *const value
XStatus	XPm_DevIoctl	const u32 deviceId const pm_i_octl_i_d ioctlId const u32 arg1 const u32 arg2 u32 *const response

Functions

XPm_InitXilpm

Note

Prototype

```
XStatus XPm_InitXilpm(XIpiPsu *IpiInst);
```

Parameters

XPm_InitXilpm

Table 2: XPm_InitXilpm Arguments

Type	Name	Description
XIpiPsu *	IpiInst	Pointer to IPI driver instance

Returns

XPm_GetBootStatus

*Note***Prototype**

```
enum  
    XPmBootStatus  
XPm_GetBootStatus(void);
```

Returns

XPm_SuspendFinalize

*Note***Prototype**

```
void XPm_SuspendFinalize(void);
```

Returns

XPm_SelfSuspend

Note

Prototype

```
XStatus XPm_SelfSuspend(const enum XPmNodeId nid, const u32 latency, const u8 state, const u64 address);
```

Parameters

XPm_SelfSuspend

Table 3: XPm_SelfSuspend Arguments

Type	Name	Description
const enum XPmNodeId	nid	Node ID of the CPU node to be suspended.
const u32	latency	Maximum wake-up latency requirement in us(microsecs)
const u8	state	Instead of specifying a maximum latency, a CPU can also explicitly request a certain power state.
const u64	address	Address from which to resume when woken up.

Returns

XPm_SetConfiguration

XPm_InitFinalize

Note

Prototype

```
XStatus XPm_InitFinalize(void);
```

Returns

XPm_RequestSuspend

Note

Prototype

```
XStatus XPm_RequestSuspend(const enum XPmNodeId target, const enum
XPmRequestAck ack, const u32 latency, const u8 state);
```

Parameters

XPm_RequestSuspend

Table 5: XPm_RequestSuspend Arguments

Type	Name	Description
const enum XPmNodeId	target	Node ID of the PU node to be suspended
const enum XPmRequestAck	ack	Requested acknowledge type
const u32	latency	Maximum wake-up latency requirement in us(micro sec)
const u8	state	Instead of specifying a maximum latency, a PU can also explicitly request a certain power state.

Returns

XPm_RequestWakeUp

Note

Prototype

```
XStatus XPm_RequestWakeUp(const enum XPmNodeId target, const bool
setAddress, const u64 address, const enum XPmRequestAck ack);
```

Parameters

XPm_RequestWakeUp

Table 6: XPm_RequestWakeUp Arguments

Type	Name	Description
const enum XPmNodeId	target	Node ID of the CPU or PU to be powered/woke up.
const bool	setAddress	Specifies whether the start address argument is being passed. <ul style="list-style-type: none"> 0 : do not set start address 1 : set start address
const u64	address	Address from which to resume when woken up. Will only be used if set_address is 1.
const enum XPmRequestAck	ack	Requested acknowledge type

Returns

XPm_ForcePowerDown

Note

Prototype

```
XStatus XPm_ForcePowerDown(const enum XPmNodeId target, const enum
XPmRequestAck ack);
```

Parameters

XPm_ForcePowerDown

Table 7: XPm_ForcePowerDown Arguments

Type	Name	Description
const enum XPmNodeId	target	Node ID of the PU node or power island/domain to be powered down.
const enum XPmRequestAck	ack	Requested acknowledge type

Returns

XPm_AbortSuspend

Note

Prototype

```
XStatus XPm_AbortSuspend(const enum XPmAbortReason reason);
```

Parameters

XPm_AbortSuspend

Table 8: **XPm_AbortSuspend Arguments**

Type	Name	Description
const enum XPmAbortReason	reason	Reason code why the suspend can not be performed or completed <ul style="list-style-type: none">ABORT_REASON_WKUP_EVENT : local wakeup-event receivedABORT_REASON_PU_BUSY : PU is busyABORT_REASON_NO_PWRDN : no external powerdown supportedABORT_REASON_UNKNOWN : unknown error during suspend procedure

Returns

XPm_SetWakeUpSource

Note

Returns

XPm_SystemShutdown

Note

Prototype

```
XStatus XPm_SystemShutdown(u32 type, u32 subtype);
```

Parameters

XPm_SystemShutdown

Table 10: XPm_SystemShutdown Arguments

Type	Name	Description
u32	type	Should the system be restarted automatically? <ul style="list-style-type: none"> PM_SHUTDOWN : no restart requested, system will be powered off permanently PM_RESTART : restart is requested, system will go through a full reset
u32	subtype	Restart subtype (SYSTEM or PS_ONLY or SUBSYSTEM)

Returns

XPm_RequestNode

Note

Prototype

```
XStatus XPm_RequestNode(const enum XPmNodeId node, const u32 capabilities,
const u32 qos, const enum XPmRequestAck ack);
```

Parameters

XPm_RequestNode

Table 11: XPm_RequestNode Arguments

Type	Name	Description
const enum XPmNodeId	node	Node ID of the PM slave requested
const u32	capabilities	Slave-specific capabilities required, can be combined <ul style="list-style-type: none"> PM_CAP_ACCESS : full access / functionality PM_CAP_CONTEXT : preserve context PM_CAP_WAKEUP : emit wake interrupts
const u32	qos	Quality of Service (0-100) required
const enum XPmRequestAck	ack	Requested acknowledge type

Returns

XPm_SetRequirement

Note

Prototype

```
XStatus XPm_SetRequirement(const enum XPmNodeId nid, const u32
capabilities, const u32 qos, const enum XPmRequestAck ack);
```

Parameters

XPm_SetRequirement

Table 12: XPm_SetRequirement Arguments

Type	Name	Description
const enum XPmNodeID	nid	Node ID of the PM slave.
const u32	capabilities	Slave-specific capabilities required.
const u32	qos	Quality of Service (0-100) required.
const enum XPmRequestAck	ack	Requested acknowledge type

Returns

XPm_ReleaseNode

Note

Prototype

```
XStatus XPm_ReleaseNode(const enum XPmNodeID node);
```

Parameters

XPm_ReleaseNode

Table 13: XPm_ReleaseNode Arguments

Type	Name	Description
const enum XPmNodeID	node	Node ID of the PM slave.

Returns

XPm_SetMaxLatency

Note

Prototype

```
XStatus XPm_SetMaxLatency(const enum XPmNodeId node, const u32 latency);
```

Parameters

XPm_SetMaxLatency

Table 14: XPm_SetMaxLatency Arguments

Type	Name	Description
const enum XPmNodeId	node	Node ID of the PM slave.
const u32	latency	Maximum wake-up latency required.

Returns

XPm_InitSuspendCb

Note

Prototype

```
void XPm_InitSuspendCb(const enum XPmSuspendReason reason, const u32 latency, const u32 state, const u32 timeout);
```

Parameters

XPm_InitSuspendCb

Table 15: XPm_InitSuspendCb Arguments

Type	Name	Description
const enum XPmSuspendReason	reason	Suspend reason: <ul style="list-style-type: none"> SUSPEND_REASON_PU_REQ : Request by another PU SUSPEND_REASON_ALERT : Unrecoverable SysMon alert SUSPEND_REASON_SHUTDOWN : System shutdown SUSPEND_REASON_RESTART : System restart

Table 15: **XPm_InitSuspendCb Arguments** (*cont'd*)

Type	Name	Description
const u32	latency	Maximum wake-up latency in us(micro secs). This information can be used by the PU to decide what level of context saving may be required.
const u32	state	Targeted sleep/suspend state.
const u32	timeout	Timeout in ms, specifying how much time a PU has to initiate its suspend procedure before it's being considered unresponsive.

XPm_NotifyCb

Note

Prototype

```
void XPm_NotifyCb(const enum XPmNodeId node, const enum XPmNotifyEvent
event, const u32 oppoint);
```

Parameters

XPm_NotifyCb

Table 17: XPm_NotifyCb Arguments

Type	Name	Description
const enum XPmNodeId	node	ID of the node the event notification is related to.
const enum XPmNotifyEvent	event	ID of the event
const u32	oppoint	Current operating state of the node.

Returns

XPm_GetApiVersion

Note

Prototype

```
XStatus XPm_GetApiVersion(u32 *version);
```

Parameters

XPm_GetApiVersion

Table 18: XPm_GetApiVersion Arguments

Type	Name	Description
u32 *	version	Returns the API 32-bit version number. Returns 0 if no PM firmware present.

Returns

XPm_GetNodeStatus

[XPm_NodeStatus](#)

Note

Prototype

```
XStatus XPm_GetNodeStatus(const enum XPmModel d_node, XPm_NodeStatus *const
nodestatus);
```

Parameters

XPm_GetNodeStatus

Table 19: XPm_GetNodeStatus Arguments

Type	Name	Description
const enum XPmModel d	node	ID of the component or sub-system in question.
XPm_NodeStatus *const	nodestatus	Used to return the complete status of the node.

Returns

XPm_GetOpCharacteristic

Note

Table 20: XPm_GetOpCharacteristic Arguments (cont'd)

Type	Name	Description
const enum XPmOpChar Type	type	Type of operating characteristic requested: <ul style="list-style-type: none"> power (current power consumption), latency (current latency in micro seconds to return to active state), temperature (current temperature),
u32 *const	result	Used to return the requested operating characteristic.

Returns

XPm_ResetAssert

Note

Prototype

```
XStatus XPm_ResetAssert(const enum XPmReset reset, const enum
XPmResetAction resetaction);
```

Parameters

XPm_ResetAssert

Table 21: XPm_ResetAssert Arguments

Type	Name	Description
const enum XPmReset	reset	ID of the reset line
const enum XPmResetAction	resetaction	Identifies action: <ul style="list-style-type: none"> PM_RESET_ACTION_RELEASE : release reset, PM_RESET_ACTION_ASSERT : assert reset, PM_RESET_ACTION_PULSE : pulse reset,

Returns

XPm_ResetGetStatus

Note

Prototype

```
XStatus XPm_ResetGetStatus(const enum XPmReset reset, u32 *status);
```

Parameters

XPm_ResetGetStatus

Table 22: XPm_ResetGetStatus Arguments

Type	Name	Description
const enum XPmReset	reset	Reset line
u32 *	status	Status of specified reset (true - asserted, false - released)

Returns

XPm_RegisterNotifier

Note

Prototype

```
XStatus XPm_RegisterNotifier(XPm_Notifier *const notifier);
```

Parameters

XPm_RegisterNotifier

Table 23: XPm_RegisterNotifier Arguments

Type	Name	Description
XPm_Notifier *const	notifier	Pointer to the notifier object to be associated with the requested notification. The notifier object contains the following data related to the notification:

Returns

XPm_UnregisterNotifier

Note

Prototype

```
XStatus XPm_UnregisterNotifier(XPm_Notifier *const notifier);
```

Parameters

XPm_UnregisterNotifier

Table 24: XPm_UnregisterNotifier Arguments

Type	Name	Description
XPm_Notifier *const	notifier	Pointer to the notifier object associated with the previously requested notification

Returns

XPm_MmioWrite

Note

Prototype

```
XStatus XPm_MmioWrite(const u32 address, const u32 mask, const u32 value);
```

Parameters

XPm_MmioWrite

Table 25: XPm_MmioWrite Arguments

Type	Name	Description
const u32	address	Physical 32-bit address of memory mapped register to write to.
const u32	mask	32-bit value used to limit write to specific bits in the register.
const u32	value	Value to write to the register bits specified by the mask.

Returns

XPm_MmioRead

Note

Prototype

```
XStatus XPm_MmioRead(const u32 address, u32 *const value);
```

Parameters

XPm_MmioRead

Table 26: XPm_MmioRead Arguments

Type	Name	Description
const u32	address	Physical 32-bit address of memory mapped register to read from.

Table 26: **XPm_MmioRead Arguments** (*cont'd*)

Type	Name	Description
u32 *const	value	Returns the 32-bit value read from the register

Returns

XPm_ClockEnable

Table 28: XPm_ClockDisable Arguments

Type	Name	Description
const enum XPmClock	clk	Identifier of the target clock to be disabled

Returns

XPm_ClockGetStatus

Prototype

```
XStatus XPm_ClockGetStatus(const enum XPmClock clk, u32 *const status);
```

Parameters

XPm_ClockGetStatus

Table 29: XPm_ClockGetStatus Arguments

Type	Name	Description
const enum XPmClock	clk	Identifier of the target clock
u32 *const	status	Location to store clock gate state (1=enabled, 0=disabled)

Returns

XPm_ClockSetOneDivider

Note

Prototype

```
XStatus XPm_ClockSetOneDivider(const enum XPmClock clk, const u32 divider, const u32 divld);
```

Parameters

XPm_ClockSetOneDivider

Table 30: Xpm_ClockSetOneDivider Arguments

Type	Name	Description
const enum XpmClock	clk	Identifier of the target clock
const u32	divider	Divider value to be set
const u32	divId	ID of the divider to be set

Returns

Xpm_ClockSetDivider

Note

Prototype

```
XStatus Xpm_ClockSetDivider(const enum XpmClock clk, const u32 divider);
```

Parameters

Xpm_ClockSetDivider

Table 31: Xpm_ClockSetDivider Arguments

Type	Name	Description
const enum XpmClock	clk	Identifier of the target clock
const u32	divider	Divider value to be set

Returns

Xpm_ClockGetOneDivider

Prototype

```
XStatus Xpm_ClockGetOneDivider(const enum XpmClock clk, u32 *const divider, const u32 divId);
```

Parameters

XPm_ClockGetOneDivider

Table 32: XPm_ClockGetOneDivider Arguments

Type	Name	Description
const enum XPmClock	clk	Identifier of the target clock
u32 *const	divider	Location to store the divider value
const u32	divId	ID of the divider

Returns

XPm_ClockGetDivider

Prototype

```
XStatus XPm_ClockGetDivider(const enum XPmClock clk, u32 *const divider);
```

Parameters

XPm_ClockGetDivider

Table 33: XPm_ClockGetDivider Arguments

Type	Name	Description
const enum XPmClock	clk	Identifier of the target clock
u32 *const	divider	Location to store the divider value

Returns

XPm_ClockSetParent

Note

Prototype

```
XStatus XPm_ClockSetParent(const enum XPmClock clk, const enum XPmClock parent);
```

Parameters

XPm_ClockSetParent

Table 34: XPm_ClockSetParent Arguments

Type	Name	Description
const enum XPmClock	clk	Identifier of the target clock
const enum XPmClock	parent	Identifier of the target parent clock

Returns

XPm_ClockGetParent

Prototype

```
XStatus XPm_ClockGetParent(const enum XPmClock clk, enum XPmClock *const parent);
```

Parameters

XPm_ClockGetParent

Table 35: XPm_ClockGetParent Arguments

Type	Name	Description
const enum XPmClock	clk	Identifier of the target clock
enum XPmClock *const	parent	Location to store clock parent ID

Returns

XPm_ClockSetRate

Note

Prototype

```
XStatus XPm_ClockSetRate(const enum XPmClock clk, const u32 rate);
```

Parameters

XPm_ClockSetRate

Table 36: XPm_ClockSetRate Arguments

Type	Name	Description
const enum XPmClock	clk	Identifier of the target clock
const u32	rate	Clock frequency (rate) to be set

Note

Prototype

```
XStatus XPm_PllSetParameter(const enum XPmNodeId node, const enum
XPmPllParameter, const u32 value);
```

Parameters

XPm_PllSetParameter

Table 38: XPm_PllSetParameter Arguments

Type	Name	Description
const enum XPmNodeId	node	PLL node identifier
const enum XPmPllParameter	parameter	PLL parameter identifier
const u32	value	Value of the PLL parameter

Returns

XPm_PllGetParameter

Prototype

```
XStatus XPm_PllGetParameter(const enum XPmNodeId node, const enum
XPmPllParameter, u32 *const value);
```

Parameters

XPm_PllGetParameter

Table 39: XPm_PllGetParameter Arguments

Type	Name	Description
const enum XPmNodeId	node	PLL node identifier
const enum XPmPllParameter	parameter	PLL parameter identifier
u32 *const	value	Location to store value of the PLL parameter

Returns

XPm_PllSetMode

Note

Prototype

```
XStatus XPm_PllSetMode(const enum XPmNodeId node, const enum XPmPllMode mode);
```

Parameters

XPm_PllSetMode

Table 40: XPm_PllSetMode Arguments

Type	Name	Description
const enum XPmNodeId	node	PLL node identifier
const enum XPmPllMode	mode	PLL mode to be set

Returns

XPm_PllGetMode

Prototype

```
XStatus XPm_PllGetMode(const enum XPmNodeId node, enum XPmPllMode *const mode);
```

Parameters

XPm_PllGetMode

Table 41: XPm_PllGetMode Arguments

Type	Name	Description
const enum XPmNodeId	node	PLL node identifier
enum XPmPllMode *const	mode	Location to store the PLL mode

Returns

XPm_PinCtrlAction

Prototype

```
XStatus XPm_Pi nCtrl Acti on(const u32 pi n, const enum XPmApi I d api );
```

Parameters

XPm_Pi nCtrl Acti on

Table 42: XPm_PinCtrlAction Arguments

Type	Name	Description
const u32	pin	PIN identifier (index from range 0-77)
const enum XPmApi I d	api	API identifier (request or release pin control)

Returns

XPm_PinCtrlRequest

Prototype

```
XStatus XPm_Pi nCtrl Request(const u32 pi n);
```

Parameters

XPm_Pi nCtrl Request

Table 43: XPm_PinCtrlRequest Arguments

Type	Name	Description
const u32	pin	PIN identifier (index from range 0-77)

Returns

XPm_PinCtrlRelease

Prototype

```
XStatus XPm_PinCtrlRelease(const u32 pin);
```

Parameters

XPm_PinCtrlRelease

Table 44: XPm_PinCtrlRelease Arguments

Type	Name	Description
const u32	pin	PIN identifier (index from range 0-77)

Returns

XPm_PinCtrlSetFunction

Note

Prototype

```
XStatus XPm_PinCtrlSetFunction(const u32 pin, const enum XPmPinFn fn);
```

Parameters

XPm_PinCtrlSetFunction

Table 45: XPm_PinCtrlSetFunction Arguments

Type	Name	Description
const u32	pin	Pin identifier
const enum XPmPinFn	fn	Pin function to be set

Returns

XPm_PinCtrlGetFunction

Prototype

```
XStatus XPm_PinCtrlGetFunction(const u32 pin, enum XPmPinFn *const fn);
```

Parameters

XPm_PinCtrlGetFunction

Table 46: XPm_PinCtrlGetFunction Arguments

Type	Name	Description
const u32	pin	PLL node identifier
enum XPmPinFn *const	fn	Location to store the pin function

Returns

XPm_PinCtrlSetParameter

Note

Prototype

```
XStatus XPm_PinCtrlSetParameter(const u32 pin, const enum XPmPinParam param, const u32 value);
```

Parameters

XPm_PinCtrlSetParameter

Table 47: XPm_PinCtrlSetParameter Arguments

Type	Name	Description
const u32	pin	Pin identifier
const enum XPmPinParam	param	Pin parameter identifier
const u32	value	Value of the pin parameter to set

Returns

XPm_PinCtrlGetParameter

Prototype

```
XStatus XPm_PinCtrlGetParameter(const u32 pin, const enum XPm_PinParam param, u32 *const value);
```

Parameters

XPm_PinCtrlGetParameter

Table 48: XPm_PinCtrlGetParameter Arguments

Type	Name	Description
const u32	pin	Pin identifier
const enum XPm_PinParam	param	Pin parameter identifier
u32 *const	value	Location to store value of the pin parameter

Returns

XPm_DevIoctl

Prototype

```
XStatus XPm_DevIoctl(const u32 deviceId, const pm_ioctl_id ioctlId, const u32 arg1, const u32 arg2, u32 *const response);
```

Parameters

XPm_DevIoctl

Table 49: XPm_DevIoctl Arguments

Type	Name	Description
const u32	deviceId	ID of the device

Table 49: **XPm_DevIoctl Arguments** (*cont*

Table 50: Enumeration XPmApiId Values (cont'd)

Value	Description
PM_MMIO_READ	0x14
PM_INIT_FINALIZE	0x15
PM_FPGA_LOAD	0x16
PM_FPGA_GET_STATUS	0x17
PM_GET_CHIPID	0x18
PM_SECURE_SHA	0x1A
PM_SECURE_RSA	0x1B
PM_PINCTRL_REQUEST	0x1C
PM_PINCTRL_RELEASE	0x1D
PM_PINCTRL_GET_FUNCTION	0x1E
PM_PINCTRL_SET_FUNCTION	0x1F
PM_PINCTRL_CONFIG_PARAM_GET	0x20
PM_PINCTRL_CONFIG_PARAM_SET	0x21
PM_IOCTL	0x22
PM_QUERY_DATA	0x23
PM_CLOCK_ENABLE	0x24
PM_CLOCK_DISABLE	0x25
PM_CLOCK_GETSTATE	0x26
PM_CLOCK_SETDIVIDER	0x27
PM_CLOCK_GETDIVIDER	0x28
PM_CLOCK_SETRATE	0x29
PM_CLOCK_GETRATE	0x2A
PM_CLOCK_SETPARENT	0x2B
PM_CLOCK_GETPARENT	0x2C
PM_SECURE_IMAGE	0x2D
PM_FPGA_READ	0x2E
PM_SECURE_AES	0x2F
PM_PLL_SET_PARAMETER	0x30
PM_PLL_GET_PARAMETER	0x31
PM_PLL_SET_MODE	0x32
PM_PLL_GET_MODE	0x33
PM_REGISTER_ACCESS	0x34
PM_EFUSE_ACCESS	0x35
PM_API_MAX	0x36

Enumeration XPmApiCbId

Table 51: Enumeration XPmApiCbId Values

Value	Description
PM_INIT_SUSPEND_CB	Suspend callback
PM_ACKNOWLEDGE_CB	Acknowledge callback
PM_NOTIFY_CB	Notify callback
PM_NOTIFY_STL_NO_OP	Notify STL No OP

Enumeration XPmNodeId

Table 52: Enumeration XPmNodeId Values

Value	Description
NODE_UNKNOWN	0x0
NODE_APU	0x1
NODE_APU_0	0x2
NODE_APU_1	0x3
NODE_APU_2	0x4
NODE_APU_3	0x5
NODE_RPU	0x6
NODE_RPU_0	0x7
NODE_RPU_1	0x8
NODE_PLD	0x9
NODE_FPD	0xA
NODE_OCM_BANK_0	0xB
NODE_OCM_BANK_1	0xC
NODE_OCM_BANK_2	0xD
NODE_OCM_BANK_3	0xE
NODE_TCM_0_A	0xF
NODE_TCM_0_B	0x10
NODE_TCM_1_A	0x11
NODE_TCM_1_B	0x12
NODE_L2	0x13
NODE_GPU_PP_0	0x14
NODE_GPU_PP_1	0x15
NODE_USB_0	0x16
NODE_USB_1	0x17
NODE_TTC_0	0x18
NODE_TTC_1	0x19
NODE_TTC_2	0x1A

Table 52: Enumeration XPmNodeId Values (cont'd)

Value	Description
NODE_TTC_3	0x1B
NODE_SATA	0x1C
NODE_ETH_0	0x1D
NODE_ETH_1	0x1E
NODE_ETH_2	0x1F
NODE_ETH_3	0x20
NODE_UART_0	0x21
NODE_UART_1	0x22
NODE_SPI_0	0x23
NODE_SPI_1	0x24
NODE_I2C_0	0x25
NODE_I2C_1	0x26
NODE_SD_0	0x27
NODE_SD_1	0x28
NODE_DP	0x29
NODE_GDMA	0x2A
NODE_ADMA	0x2B
NODE_NAND	0x2C
NODE_QSPI	0x2D
NODE_GPIO	0x2E
NODE_CAN_0	0x2F
NODE_CAN_1	0x30
NODE_EXTERN	0x31
NODE_APLL	0x32
NODE_VPLL	0x33
NODE_DPLL	0x34
NODE_RPLL	0x35
NODE_IOPLL	0x36
NODE_DDR	0x37
NODE_IPI_APU	0x38
NODE_IPI_RPU_0	0x39
NODE_GPU	0x3A
NODE_PCIE	0x3B
NODE_PCAP	0x3C
NODE_RTC	0x3D
NODE_LPD	0x3E
NODE_VCU	0x3F
NODE_IPI_RPU_1	0x40
NODE_IPI_PL_0	0x41

Table 52: Enumeration XPmNodeId Values (cont'd)

Value	Description
NODE_IPI_PL_1	0x42
NODE_IPI_PL_2	0x43
NODE_IPI_PL_3	0x44
NODE_PL	0x45
NODE_ID_MAX	0x46

Enumeration XPmRequestAck

Table 53: Enumeration XPmRequestAck Values

Value	Description
REQUEST_ACK_NO	No Ack
REQUEST_ACK_BLOCKING	Blocking Ack
REQUEST_ACK_NON_BLOCKING	Non blocking Ack
REQUEST_ACK_CB_CERROR	Callback Error

Enumeration XPmAbortReason

Table 54: Enumeration XPmAbortReason Values

Value	Description
ABORT_REASON_WKUP_EVENT	Wakeup Event
ABORT_REASON_PU_BUSY	Processor Busy
ABORT_REASON_NO_PWRDN	No Powerdown
ABORT_REASON_UNKNOWN	Unknown Reason

Enumeration XPmSuspendReason

Table 55: Enumeration XPmSuspendReason Values

Value	Description
SUSPEND_REASON_PU_REQ	Processor request
SUSPEND_REASON_ALERT	Alert
SUSPEND_REASON_SYS_SHUTDOWN	System shutdown

Enumeration XPmRamState

Table 56: Enumeration XPmRamState Values

Value	Description
PM_RAM_STATE_OFF	Off
PM_RAM_STATE_RETENTION	Retention
PM_RAM_STATE_ON	On

Enumeration XPmOpChar p

Description**Define PM_VERSION****Definition**

```
#define PM_VERSION((  
    PM_VERSION_MAJOR  
    << 16) |  
    PM_VERSION_MINOR  
)
```

Description**Define PM_CAP_ACCESS****Definition**

```
#define PM_CAP_ACCESS0x1U
```

Description**Define PM_CAP_CONTEXT****Definition**

```
#define PM_CAP_CONTEXT0x2U
```

Description**Define PM_CAP_WAKEUP****Definition**

```
#define PM_CAP_WAKEUP0x4U
```

Description**Define NODE_STATE_OFF****Definition**

```
#define NODE_STATE_OFF 0
```

Description**Define NODE_STATE_ON****Definition**

```
#define NODE_STATE_ON 1
```

Description**Define PROC_STATE_FORCEDOFF****Definition**

```
#define PROC_STATE_FORCEDOFF 0
```

Description**Define PROC_STATE_ACTIVE****Definition**

```
#define PROC_STATE_ACTIVE 1
```

Description

Define PROC_STATE_SLEEP

Definition

```
#define PROC_STATE_SLEEP2
```

Description

Define PROC_STATE_SUSPENDING

Definition

```
#define PROC_STATE_SUSPENDING3
```

Description

Define MAX_LATENCY

Definition

```
#define MAX_LATENCY(~0U)
```

Description

Define MAX_QOS

Definition

```
#define MAX_QOS100U
```

Description

Define PMF_SHUTDOWN_TYPE_SHUTDOWN

Definition

```
#define PMF_SHUTDOWN_TYPE_SHUTDOWN0U
```

Description

Define PMF_SHUTDOWN_TYPE_RESET

Definition

```
#define PMF_SHUTDOWN_TYPE_RESET1U
```

Description

Define PMF_SHUTDOWN_SUBTYPE_SUBSYSTEM

Definition

```
#define PMF_SHUTDOWN_SUBTYPE_SUBSYSTEM0U
```

Description

Define PMF_SHUTDOWN_SUBTYPE_PS_ONLY

Definition

```
#define PMF_SHUTDOWN_SUBTYPE_PS_ONLY1U
```

Description

Define PMF_SHUTDOWN_SUBTYPE_SYSTEM

Definition

```
#define PMF_SHUTDOWN_SUBTYPE_SYSTEM2U
```

Description

Table 60: Enumeration XPmReset Values

Value	Description
XILPM_RESET_PCIE_CFG	Reset ID PCIE_CFG
XILPM_RESET_PCIE_BRIDGE	Reset ID PCIE_BRIDGE
XILPM_RESET_PCIE_CTRL	Reset ID PCIE_CTRL
XILPM_RESET_DP	Reset ID DP
XILPM_RESET_SWDT_CRF	Reset ID SWDT_CRF
XILPM_RESET_AFI_FM5	Reset ID AFI_FM5
XILPM_RESET_AFI_FM4	Reset ID AFI_FM4
XILPM_RESET_AFI_FM3	Reset ID AFI_FM3
XILPM_RESET_AFI_FM2	Reset ID AFI_FM2
XILPM_RESET_AFI_FM1	Reset ID AFI_FM1
XILPM_RESET_AFI_FM0	Reset ID AFI_FM0
XILPM_RESET_GDMA	Reset ID GDMA
XILPM_RESET_GPU_PP1	Reset ID GPU_PP1
XILPM_RESET_GPU_PP0	Reset ID GPU_PP0
XILPM_RESET_GPU	Reset ID GPU
XILPM_RESET_GT	Reset ID GT
XILPM_RESET_SATA	Reset ID SATA
XILPM_RESET_ACPU3_PWRON	Reset ID ACPU3_PWRON
XILPM_RESET_ACPU2_PWRON	Reset ID ACPU2_PWRON
XILPM_RESET_ACPU1_PWRON	Reset ID ACPU1_PWRON
XILPM_RESET_ACPU0_PWRON	Reset ID ACPU0_PWRON
XILPM_RESET_APU_L2	Reset ID APU_L2
XILPM_RESET_ACPU3	Reset ID ACPU3
XILPM_RESET_ACPU2	Reset ID ACPU2
XILPM_RESET_ACPU1	Reset ID ACPU1
XILPM_RESET_ACPU0	Reset ID ACPU0
XILPM_RESET_DDR	Reset ID DDR
XILPM_RESET_APM_FPD	Reset ID APM_FPD
XILPM_RESET_SOFT	Reset ID SOFT
XILPM_RESET_GEM0	Reset ID GEM0
XILPM_RESET_GEM1	Reset ID GEM1
XILPM_RESET_GEM2	Reset ID GEM2
XILPM_RESET_GEM3	Reset ID GEM3
XILPM_RESET_QSPI	Reset ID QSPI
XILPM_RESET_UART0	Reset ID UART0
XILPM_RESET_UART1	Reset ID UART1
XILPM_RESET_SPI0	Reset ID SPI0
XILPM_RESET_SPI1	Reset ID SPI1
XILPM_RESET_SDIO0	Reset ID SDIO0

Table 60: Enumeration XPmReset Values (cont'd)

Value	Description
XILPM_RESET_SDIO1	Reset ID SDIO1
XILPM_RESET_CAN0	Reset ID CAN0
XILPM_RESET_CAN1	Reset ID CAN1
XILPM_RESET_I2C0	Reset ID I2C0
XILPM_RESET_I2C1	Reset ID I2C1
XILPM_RESET_TTC0	Reset ID TTC0
XILPM_RESET_TTC1	Reset ID TTC1
XILPM_RESET_TTC2	Reset ID TTC2
XILPM_RESET_TTC3	Reset ID TTC3
XILPM_RESET_SWDT_CRL	Reset ID SWDT_CRL
XILPM_RESET_NAND	Reset ID NAND
XILPM_RESET_ADMA	Reset ID ADMA
XILPM_RESET_GPIO	Reset ID GPIO
XILPM_RESET_I0U_CC	Reset ID I0U_CC
XILPM_RESET_TIMESTAMP	Reset ID TIMESTAMP
XILPM_RESET_RPU_R50	Reset ID RPU_R50
XILPM_RESET_RPU_R51	Reset ID RPU_R51
XILPM_RESET_RPU_AMBA	Reset ID RPU_AMBA
XILPM_RESET_OCM	Reset ID OCM
XILPM_RESET_RPU_PGE	Reset ID RPU_PGE
XILPM_RESET_USB0_CORERESET	Reset ID USB0_CORERESE
XILPM_RESET_USB1_CORERESET	Reset ID USB1_CORERESE
XILPM_RESET_USB0_HIBERRESET	Reset ID USB0_HIBERRES
XILPM_RESET_USB1_HIBERRESET	Reset ID USB1_HIBERRES
XILPM_RESET_USB0_APB	Reset ID USB0_APB
XILPM_RESET_USB1_APB	Reset ID USB1_APB
XILPM_RESET_IPI	Reset ID IPI
XILPM_RESET_APM_LPD	Reset ID APM_LPD
XILPM_RESET_RTC	Reset ID RTC
XILPM_RESET_SYSMON	Reset ID SYSMON
XILPM_RESET_AFI_FM6	Reset ID AFI_FM6
XILPM_RESET_LPD_SWDT	Reset ID LPD_SWDT
XILPM_RESET_FPD	Reset ID FPD
XILPM_RESET_RPU_DBG1	Reset ID RPU_DBG1
XILPM_RESET_RPU_DBG0	Reset ID RPU_DBG0
XILPM_RESET_DBG_LPD	Reset ID DBG_LPD
XILPM_RESET_DBG_FPD	Reset ID DBG_FPD
XILPM_RESET_APLL	Reset ID APLL
XILPM_RESET_DPLL	Reset ID DPLL

Table 60: Enumeration XPmReset Values (cont'd)

Value	Description
XILPM_RESET_VPLL	Reset ID VPLL
XILPM_RESET_IOPLL	Reset ID IOPLL
XILPM_RESET_RPLL	Reset ID RPLL
XILPM_RESET_GPO3_PL_0	Reset ID GPO3_PL_0
XILPM_RESET_GPO3_PL_1	Reset ID GPO3_PL_1
XILPM_RESET_GPO3_PL_2	Reset ID GPO3_PL_2
XILPM_RESET_GPO3_PL_3	Reset ID GPO3_PL_3
XILPM_RESET_GPO3_PL_4	Reset ID GPO3_PL_4
XILPM_RESET_GPO3_PL_5	Reset ID GPO3_PL_5
XILPM_RESET_GPO3_PL_6	Reset ID GPO3_PL_6
XILPM_RESET_GPO3_PL_7	Reset ID GPO3_PL_7
XILPM_RESET_GPO3_PL_8	Reset ID GPO3_PL_8
XILPM_RESET_GPO3_PL_9	Reset ID GPO3_PL_9
XILPM_RESET_GPO3_PL_10	Reset ID GPO3_PL_10
XILPM_RESET_GPO3_PL_11	Reset ID GPO3_PL_11
XILPM_RESET_GPO3_PL_12	Reset ID GPO3_PL_12
XILPM_RESET_GPO3_PL_13	Reset ID GPO3_PL_13
XILPM_RESET_GPO3_PL_14	Reset ID GPO3_PL_14
XILPM_RESET_GPO3_PL_15	Reset ID GPO3_PL_15
XILPM_RESET_GPO3_PL_16	Reset ID GPO3_PL_16
XILPM_RESET_GPO3_PL_17	Reset ID GPO3_PL_17
XILPM_RESET_GPO3_PL_18	Reset ID GPO3_PL_18
XILPM_RESET_GPO3_PL_19	Reset ID GPO3_PL_19
XILPM_RESET_GPO3_PL_20	Reset ID GPO3_PL_20
XILPM_RESET_GPO3_PL_21	Reset ID GPO3_PL_21
XILPM_RESET_GPO3_PL_22	Reset ID GPO3_PL_22
XILPM_RESET_GPO3_PL_23	Reset ID GPO3_PL_23
XILPM_RESET_GPO3_PL_24	Reset ID GPO3_PL_24
XILPM_RESET_GPO3_PL_25	Reset ID GPO3_PL_25
XILPM_RESET_GPO3_PL_26	Reset ID GPO3_PL_26
XILPM_RESET_GPO3_PL_27	Reset ID GPO3_PL_27
XILPM_RESET_GPO3_PL_28	Reset ID GPO3_PL_28
XILPM_RESET_GPO3_PL_29	Reset ID GPO3_PL_29
XILPM_RESET_GPO3_PL_30	Reset ID GPO3_PL_30
XILPM_RESET_GPO3_PL_31	Reset ID GPO3_PL_31
XILPM_RESET_RPU_LS	Reset ID RPU_LS
XILPM_RESET_PS_ONLY	Reset ID PS_ONLY
XILPM_RESET_PL	Reset ID PL
XILPM_RESET_GPIO5_EMIO_92	Reset ID GPIO5_EMIO_92

Table 60: Enumeration XPmReset Values (cont'd)

Value	Description
XILPM_RESET_GPIO5_EMIO_93	Reset ID GPIO5_EMIO_93
XILPM_RESET_GPIO5_EMIO_94	Reset ID GPIO5_EMIO_94
XILPM_RESET_GPIO5_EMIO_95	Reset ID GPIO5_EMIO_95

Enumeration XPmNotifyEvent

Table 61: Enumeration XPmNotifyEvent Values

Value	Description
EVENT_STATE_CHANGE	State change event
EVENT_ZERO_USERS	Zero user event
EVENT_CPU_IDLE_FORCE_PWRDWN	CPU idle event during force power down

Enumeration XPmClock

Table 62: Enumeration XPmClock Values

Value	Description
PM_CLOCK_IOPLL	Clock ID IOPLL
PM_CLOCK_RPLL	Clock ID RPLL
PM_CLOCK_APLL	Clock ID APLL
PM_CLOCK_DPLL	Clock ID DPLL
PM_CLOCK_VPLL	Clock ID VPLL
PM_CLOCK_IOPLL_TO_FPD	Clock ID IOPLL_TO_FPD
PM_CLOCK_RPLL_TO_FPD	Clock ID RPLL_TO_FPD
PM_CLOCK_APLL_TO_LPD	Clock ID APLL_TO_LPD
PM_CLOCK_DPLL_TO_LPD	Clock ID DPLL_TO_LPD
PM_CLOCK_VPLL_TO_LPD	Clock ID VPLL_TO_LPD
PM_CLOCK_ACPU	Clock ID ACPU
PM_CLOCK_ACPU_HALF	Clock ID ACPU_HALF
PM_CLOCK_DBG_FPD	Clock ID DBG_FPD
PM_CLOCK_DBG_LPD	Clock ID DBG_LPD
PM_CLOCK_DBG_TRACE	Clock ID DBG_TRACE
PM_CLOCK_DBG_TSTMP	Clock ID DBG_TSTMP
PM_CLOCK_DP_VIDEO_REF	Clock ID DP_VIDEO_REF
PM_CLOCK_DP_AUDIO_REF	Clock ID DP_AUDIO_REF

Table 62: Enumeration XPmClock Values (cont'd)

Value	Description
PM_CLOCK_DP_STC_REF	Clock ID DP_STC_REF
PM_CLOCK_GDMA_REF	Clock ID GDMA_REF
PM_CLOCK_DPDMA_REF	Clock ID DPDMA_REF
PM_CLOCK_DDR_REF	Clock ID DDR_REF
PM_CLOCK_SATA_REF	Clock ID SATA_REF
PM_CLOCK_PCIE_REF	Clock ID PCIE_REF
PM_CLOCK_GPU_REF	Clock ID GPU_REF
PM_CLOCK_GPU_PP0_REF	Clock ID GPU_PP0_REF
PM_CLOCK_GPU_PP1_REF	Clock ID GPU_PP1_REF
PM_CLOCK_TOPSW_MAIN	Clock ID TOPSW_MAIN
PM_CLOCK_TOPSW_LSBUS	Clock ID TOPSW_LSBUS
PM_CLOCK_GTGREF0_REF	Clock ID GTGREF0_REF
PM_CLOCK_LPD_SWITCH	Clock ID LPD_SWITCH
PM_CLOCK_LPD_LSBUS	Clock ID LPD_LSBUS
PM_CLOCK_USB0_BUS_REF	Clock ID USB0_BUS_REF
PM_CLOCK_USB1_BUS_REF	Clock ID USB1_BUS_REF
PM_CLOCK_USB3_DUAL_REF	Clock ID USB3_DUAL_REF
PM_CLOCK_USB0	Clock ID USB0
PM_CLOCK_USB1	Clock ID USB1
PM_CLOCK_CPU_R5	Clock ID CPU_R5
PM_CLOCK_CPU_R5_CORE	Clock ID CPU_R5_CORE
PM_CLOCK_CSU_SPB	Clock ID CSU_SPB
PM_CLOCK_CSU_PLL	Clock ID CSU_PLL
PM_CLOCK_PCAP	Clock ID PCAP
PM_CLOCK_IOW_SWITCH	Clock ID IOW_SWITCH
PM_CLOCK_GEM_TSU_REF	Clock ID GEM_TSU_REF
PM_CLOCK_GEM_TSU	Clock ID GEM_TSU
PM_CLOCK_GEM0_TX	Clock ID GEM0_TX
PM_CLOCK_GEM1_TX	Clock ID GEM1_TX
PM_CLOCK_GEM2_TX	Clock ID GEM2_TX
PM_CLOCK_GEM3_TX	Clock ID GEM3_TX
PM_CLOCK_GEM0_RX	Clock ID GEM0_RX
PM_CLOCK_GEM1_RX	Clock ID GEM1_RX
PM_CLOCK_GEM2_RX	Clock ID GEM2_RX
PM_CLOCK_GEM3_RX	Clock ID GEM3_RX
PM_CLOCK_QSPI_REF	Clock ID QSPI_REF
PM_CLOCK_SDIO0_REF	Clock ID SDIO0_REF
PM_CLOCK_SDIO1_REF	Clock ID SDIO1_REF
PM_CLOCK_UART0_REF	Clock ID UART0_REF

Table 62: Enumeration XPmClock Values (cont'd)

Value	Description
PM_CLOCK_UART1_REF	Clock ID UART1_REF
PM_CLOCK_SPI0_REF	Clock ID SPI0_REF
PM_CLOCK_SPI1_REF	Clock ID SPI1_REF
PM_CLOCK_NAND_REF	Clock ID NAND_REF
PM_CLOCK_I2C0_REF	Clock ID I2C0_REF
PM_CLOCK_I2C1_REF	Clock ID I2C1_REF
PM_CLOCK_CAN0_REF	Clock ID CAN0_REF
PM_CLOCK_CAN1_REF	Clock ID CAN1_REF
PM_CLOCK_CAN0	Clock ID CAN0
PM_CLOCK_CAN1	Clock ID CAN1
PM_CLOCK_DLL_REF	Clock ID DLL_REF
PM_CLOCK_ADMA_REF	Clock ID ADMA_REF
PM_CLOCK_TIMESTAMP_REF	Clock ID TIMESTAMP_REF
PM_CLOCK_AMS_REF	Clock ID AMS_REF
PM_CLOCK_PL0_REF	Clock ID PL0_REF
PM_CLOCK_PL1_REF	Clock ID PL1_REF
PM_CLOCK_PL2_REF	Clock ID PL2_REF
PM_CLOCK_PL3_REF	Clock ID PL3_REF
PM_CLOCK_WDT	Clock ID WDT
PM_CLOCK_IOPLL_INT	Clock ID IOPLL_INT
PM_CLOCK_IOPLL_PRE_SRC	Clock ID IOPLL_PRE_SRC
PM_CLOCK_IOPLL_HALF	Clock ID IOPLL_HALF
PM_CLOCK_IOPLL_INT_MUX	Clock ID IOPLL_INT_MUX
PM_CLOCK_IOPLL_POST_SRC	Clock ID IOPLL_POST_SRC
PM_CLOCK_RPLL_INT	Clock ID RPLL_INT
PM_CLOCK_RPLL_PRE_SRC	Clock ID RPLL_PRE_SRC
PM_CLOCK_RPLL_HALF	Clock ID RPLL_HALF
PM_CLOCK_RPLL_INT_MUX	Clock ID RPLL_INT_MUX
PM_CLOCK_RPLL_POST_SRC	Clock ID RPLL_POST_SRC
PM_CLOCK_APLL_INT	Clock ID APLL_INT
PM_CLOCK_APLL_PRE_SRC	Clock ID APLL_PRE_SRC
PM_CLOCK_APLL_HALF	Clock ID APLL_HALF
PM_CLOCK_APLL_INT_MUX	Clock ID APLL_INT_MUX
PM_CLOCK_APLL_POST_SRC	Clock ID APLL_POST_SRC
PM_CLOCK_DPLL_INT	Clock ID DPLL_INT
PM_CLOCK_DPLL_PRE_SRC	Clock ID DPLL_PRE_SRC
PM_CLOCK_DPLL_HALF	Clock ID DPLL_HALF
PM_CLOCK_DPLL_INT_MUX	Clock ID DPLL_INT_MUX
PM_CLOCK_DPLL_POST_SRC	Clock ID DPLL_POST_SRC

Table 62: Enumeration XPmClock Values (cont'd)

Value	Description
PM_CLOCK_VPLL_INT	Clock ID VPLL_INT
PM_CLOCK_VPLL_PRE_SRC	Clock ID VPLL_PRE_SRC
PM_CLOCK_VPLL_HALF	Clock ID VPLL_HALF
PM_CLOCK_VPLL_INT_MUX	Clock ID VPLL_INT_MUX
PM_CLOCK_VPLL_POST_SRC	Clock ID VPLL_POST_SRC
PM_CLOCK_CAN0_MIO	Clock ID CAN0_MIO
PM_CLOCK_CAN1_MIO	Clock ID CAN1_MIO
PM_CLOCK_ACPU_FULL	Clock ID ACPU_FULL
PM_CLOCK_GEM0_REF	Clock ID GEM0_REF
PM_CLOCK_GEM1_REF	Clock ID GEM1_REF
PM_CLOCK_GEM2_REF	Clock ID GEM2_REF
PM_CLOCK_GEM3_REF	Clock ID GEM3_REF
PM_CLOCK_GEM0_REF_UNGATED	Clock ID GEM0_REF_UNGATED
PM_CLOCK_GEM1_REF_UNGATED	Clock ID GEM1_REF_UNGATED
PM_CLOCK_GEM2_REF_UNGATED	Clock ID GEM2_REF_UNGATED
PM_CLOCK_GEM3_REF_UNGATED	Clock ID GEM3_REF_UNGATED
PM_CLOCK_EXT_PSS_REF	Clock ID EXT_PSS_REF
PM_CLOCK_EXT_VIDEO	Clock ID EXT_VIDEO
PM_CLOCK_EXT_PSS_ALT_REF	Clock ID EXT_PSS_ALT_REF
PM_CLOCK_EXT_AUX_REF	Clock ID EXT_AUX_REF
PM_CLOCK_EXT_GT_CRX_REF	Clock ID EXT_GT_CRX_REF
PM_CLOCK_EXT_SWDT0	Clock ID EXT_SWDT0
PM_CLOCK_EXT_SWDT1	Clock ID EXT_SWDT1
PM_CLOCK_EXT_GEM0_TX_EMIO	Clock ID EXT_GEM0_TX_EMIO
PM_CLOCK_EXT_GEM1_TX_EMIO	Clock ID EXT_GEM1_TX_EMIO
PM_CLOCK_EXT_GEM2_TX_EMIO	Clock ID EXT_GEM2_TX_EMIO
PM_CLOCK_EXT_GEM3_TX_EMIO	Clock ID EXT_GEM3_TX_EMIO
PM_CLOCK_EXT_GEM0_RX_EMIO	Clock ID EXT_GEM0_RX_EMIO
PM_CLOCK_EXT_GEM1_RX_EMIO	Clock ID EXT_GEM1_RX_EMIO
PM_CLOCK_EXT_GEM2_RX_EMIO	Clock ID EXT_GEM2_RX_EMIO
PM_CLOCK_EXT_GEM3_RX_EMIO	Clock ID EXT_GEM3_RX_EMIO
PM_CLOCK_EXT_MIO50_OR_MIO51	Clock ID EXT_MIO50_OR_MIO51
PM_CLOCK_EXT_MIO0	Clock ID EXT_MIO0
PM_CLOCK_EXT_MIO1	Clock ID EXT_MIO1
PM_CLOCK_EXT_MIO2	Clock ID EXT_MIO2
PM_CLOCK_EXT_MIO3	Clock ID EXT_MIO3
PM_CLOCK_EXT_MIO4	Clock ID EXT_MIO4
PM_CLOCK_EXT_MIO5	Clock ID EXT_MIO5
PM_CLOCK_EXT_MIO6	Clock ID EXT_MIO6

Table 62: Enumeration XPmClock Values (cont'd)

Value	Description
PM_CLOCK_EXT_MIO7	Clock ID EXT_MIO7
PM_CLOCK_EXT_MIO8	Clock ID EXT_MIO8
PM_CLOCK_EXT_MIO9	Clock ID EXT_MIO9
PM_CLOCK_EXT_MIO10	Clock ID EXT_MIO10
PM_CLOCK_EXT_MIO11	Clock ID EXT_MIO11
PM_CLOCK_EXT_MIO12	Clock ID EXT_MIO12
PM_CLOCK_EXT_MIO13	Clock ID EXT_MIO13
PM_CLOCK_EXT_MIO14	Clock ID EXT_MIO14
PM_CLOCK_EXT_MIO15	Clock ID EXT_MIO15
PM_CLOCK_EXT_MIO16	Clock ID EXT_MIO16
PM_CLOCK_EXT_MIO17	Clock ID EXT_MIO17
PM_CLOCK_EXT_MIO18	Clock ID EXT_MIO18
PM_CLOCK_EXT_MIO19	Clock ID EXT_MIO19
PM_CLOCK_EXT_MIO20	Clock ID EXT_MIO20
PM_CLOCK_EXT_MIO21	Clock ID EXT_MIO21
PM_CLOCK_EXT_MIO22	Clock ID EXT_MIO22
PM_CLOCK_EXT_MIO23	Clock ID EXT_MIO23
PM_CLOCK_EXT_MIO24	Clock ID EXT_MIO24
PM_CLOCK_EXT_MIO25	Clock ID EXT_MIO25
PM_CLOCK_EXT_MIO26	Clock ID EXT_MIO26
PM_CLOCK_EXT_MIO27	Clock ID EXT_MIO27
PM_CLOCK_EXT_MIO28	Clock ID EXT_MIO28
PM_CLOCK_EXT_MIO29	Clock ID EXT_MIO29
PM_CLOCK_EXT_MIO30	Clock ID EXT_MIO30
PM_CLOCK_EXT_MIO31	Clock ID EXT_MIO31
PM_CLOCK_EXT_MIO32	Clock ID EXT_MIO32
PM_CLOCK_EXT_MIO33	Clock ID EXT_MIO33
PM_CLOCK_EXT_MIO34	Clock ID EXT_MIO34
PM_CLOCK_EXT_MIO35	Clock ID EXT_MIO35
PM_CLOCK_EXT_MIO36	Clock ID EXT_MIO36
PM_CLOCK_EXT_MIO37	Clock ID EXT_MIO37
PM_CLOCK_EXT_MIO38	Clock ID EXT_MIO38
PM_CLOCK_EXT_MIO39	Clock ID EXT_MIO39
PM_CLOCK_EXT_MIO40	Clock ID EXT_MIO40
PM_CLOCK_EXT_MIO41	Clock ID EXT_MIO41
PM_CLOCK_EXT_MIO42	Clock ID EXT_MIO42
PM_CLOCK_EXT_MIO43	Clock ID EXT_MIO43
PM_CLOCK_EXT_MIO44	Clock ID EXT_MIO44
PM_CLOCK_EXT_MIO45	Clock ID EXT_MIO45

Table 62: Enumeration XPmClock Values (cont'd)

Value	Description
PM_CLOCK_EXT_MIO46	Clock ID EXT_MIO46
PM_CLOCK_EXT_MIO47	Clock ID EXT_MIO47
PM_CLOCK_EXT_MIO48	Clock ID EXT_MIO48
PM_CLOCK_EXT_MIO49	Clock ID EXT_MIO49
PM_CLOCK_EXT_MIO50	Clock ID EXT_MIO50
PM_CLOCK_EXT_MIO51	Clock ID EXT_MIO51
PM_CLOCK_EXT_MIO52	Clock ID EXT_MIO52
PM_CLOCK_EXT_MIO53	Clock ID EXT_MIO53
PM_CLOCK_EXT_MIO54	Clock ID EXT_MIO54
PM_CLOCK_EXT_MIO55	Clock ID EXT_MIO55
PM_CLOCK_EXT_MIO56	Clock ID EXT_MIO56
PM_CLOCK_EXT_MIO57	Clock ID EXT_MIO57
PM_CLOCK_EXT_MIO58	Clock ID EXT_MIO58
PM_CLOCK_EXT_MIO59	Clock ID EXT_MIO59
PM_CLOCK_EXT_MIO60	Clock ID EXT_MIO60
PM_CLOCK_EXT_MIO61	Clock ID EXT_MIO61
PM_CLOCK_EXT_MIO62	Clock ID EXT_MIO62
PM_CLOCK_EXT_MIO63	Clock ID EXT_MIO63
PM_CLOCK_EXT_MIO64	Clock ID EXT_MIO64
PM_CLOCK_EXT_MIO65	Clock ID EXT_MIO65
PM_CLOCK_EXT_MIO66	Clock ID EXT_MIO66
PM_CLOCK_EXT_MIO67	Clock ID EXT_MIO67
PM_CLOCK_EXT_MIO68	Clock ID EXT_MIO68
PM_CLOCK_EXT_MIO69	Clock ID EXT_MIO69
PM_CLOCK_EXT_MIO70	Clock ID EXT_MIO70
PM_CLOCK_EXT_MIO71	Clock ID EXT_MIO71
PM_CLOCK_EXT_MIO72	Clock ID EXT_MIO72
PM_CLOCK_EXT_MIO73	Clock ID EXT_MIO73
PM_CLOCK_EXT_MIO74	Clock ID EXT_MIO74
PM_CLOCK_EXT_MIO75	Clock ID EXT_MIO75
PM_CLOCK_EXT_MIO76	Clock ID EXT_MIO76
PM_CLOCK_EXT_MIO77	Clock ID EXT_MIO77

Enumeration XPmPllParam

Table 63: Enumeration XPmPllParam Values

Value	Description
PM_PLL_PARAM_ID_DIV2	PLL param ID DIV2
PM_PLL_PARAM_ID_FBDIV	PLL param ID FBDIV
PM_PLL_PARAM_ID_DATA	PLL param ID DATA
PM_PLL_PARAM_ID_PRE_SRC	PLL param ID PRE_SRC
PM_PLL_PARAM_ID_POST_SRC	PLL param ID POST_SRC
PM_PLL_PARAM_ID_LOCK_DLY	PLL param ID LOCK_DLY
PM_PLL_PARAM_ID_LOCK_CNT	PLL param ID LOCK_CNT
PM_PLL_PARAM_ID_LFHF	PLL param ID LFHF
PM_PLL_PARAM_ID_CP	PLL param ID CP
PM_PLL_PARAM_ID_RES	PLL param ID RES

Enumeration XPmPllMode

Table 64: Enumeration XPmPllMode Values

Value	Description
PM_PLL_MODE_INTEGER	PLL mode integer
PM_PLL_MODE_FRACTIONAL	PLL mode fractional
PM_PLL_MODE_RESET	PLL mode reset

Enumeration XPmPinFn

Table 65: Enumeration XPmPinFn Values

Value	Description
PINCTRL_FUNC_CAN0	Pin Function CAN0
PINCTRL_FUNC_CAN1	Pin Function CAN1
PINCTRL_FUNC_ETHERNET0	Pin Function ETHERNET0
PINCTRL_FUNC_ETHERNET1	Pin Function ETHERNET1
PINCTRL_FUNC_ETHERNET2	Pin Function ETHERNET2
PINCTRL_FUNC_ETHERNET3	Pin Function ETHERNET3
PINCTRL_FUNC_GEMTSU0	Pin Function GEMTSU0
PINCTRL_FUNC_GPIO0	Pin Function GPIO0
PINCTRL_FUNC_I2C0	Pin Function I2C0
PINCTRL_FUNC_I2C1	Pin Function I2C1
PINCTRL_FUNC_MDIO0	Pin Function MDIO0

Table 65: Enumeration XPmPinFn Values (cont'd)

Value	Description
PINCTRL_FUNC_MDIO1	Pin Function MDIO1
PINCTRL_FUNC_MDIO2	Pin Function MDIO2
PINCTRL_FUNC_MDIO3	Pin Function MDIO3
PINCTRL_FUNC_QSPI0	Pin Function QSPI0
PINCTRL_FUNC_QSPI_FBCLK	Pin Function QSPI_FBCLK
PINCTRL_FUNC_QSPI_SS	Pin Function QSPI_SS
PINCTRL_FUNC_SPI0	Pin Function SPI0
PINCTRL_FUNC_SPI1	Pin Function SPI1
PINCTRL_FUNC_SPI0_SS	Pin Function SPI0_SS
PINCTRL_FUNC_SPI1_SS	Pin Function SPI1_SS
PINCTRL_FUNC_SDIO0	Pin Function SDIO0
PINCTRL_FUNC_SDIO0_PC	Pin Function SDIO0_PC
PINCTRL_FUNC_SDIO0_CD	Pin Function SDIO0_CD
PINCTRL_FUNC_SDIO0_WP	Pin Function SDIO0_WP
PINCTRL_FUNC_SDIO1	Pin Function SDIO1
PINCTRL_FUNC_SDIO1_PC	Pin Function SDIO1_PC
PINCTRL_FUNC_SDIO1_CD	Pin Function SDIO1_CD
PINCTRL_FUNC_SDIO1_WP	Pin Function SDIO1_WP
PINCTRL_FUNC_NAND0	Pin Function NAND0
PINCTRL_FUNC_NAND0_CE	Pin Function NAND0_CE
PINCTRL_FUNC_NAND0_RB	Pin Function NAND0_RB
PINCTRL_FUNC_NAND0_DQS	Pin Function NAND0_DQS
PINCTRL_FUNC_TTC0_CLK	Pin Function TTC0_CLK
PINCTRL_FUNC_TTC0_WAV	Pin Function TTC0_WAV
PINCTRL_FUNC_TTC1_CLK	Pin Function TTC1_CLK
PINCTRL_FUNC_TTC1_WAV	Pin Function TTC1_WAV
PINCTRL_FUNC_TTC2_CLK	Pin Function TTC2_CLK
PINCTRL_FUNC_TTC2_WAV	Pin Function TTC2_WAV
PINCTRL_FUNC_TTC3_CLK	Pin Function TTC3_CLK
PINCTRL_FUNC_TTC3_WAV	Pin Function TTC3_WAV
PINCTRL_FUNC_UART0	Pin Function UART0
PINCTRL_FUNC_UART1	Pin Function UART1
PINCTRL_FUNC_USB0	Pin Function USB0
PINCTRL_FUNC_USB1	Pin Function USB1
PINCTRL_FUNC_SWDT0_CLK	Pin Function SWDT0_CLK
PINCTRL_FUNC_SWDT0_RST	Pin Function SWDT0_RST
PINCTRL_FUNC_SWDT1_CLK	Pin Function SWDT1_CLK
PINCTRL_FUNC_SWDT1_RST	Pin Function SWDT1_RST
PINCTRL_FUNC_PMU0	Pin Function PMU0

Table 65: Enumeration XPmPinFn Values (cont'd)

Value	Description
PINCTRL_FUNC_PCIE0	Pin Function PCIE0
PINCTRL_FUNC_CSU0	Pin Function CSU0
PINCTRL_FUNC_DPAUX0	Pin Function DPAUX0
PINCTRL_FUNC_PJTAG0	Pin Function PJTAG0
PINCTRL_FUNC_TRACE0	Pin Function TRACE0
PINCTRL_FUNC_TRACE0_CLK	Pin Function TRACE0_CLK
PINCTRL_FUNC_TESTSCAN0	Pin Function TESTSCAN0

Enumeration XPmPinParam

Table 66: Enumeration XPmPinParam Values

Value	Description
PINCTRL_CONFIG_SLEW_RATE	Pin config slew rate
PINCTRL_CONFIG_BIAS_STATUS	Pin config bias status
PINCTRL_CONFIG_PULL_CTRL	Pin config pull control
PINCTRL_CONFIG_SCHMITT_CMOS	Pin config schmitt CMOS
PINCTRL_CONFIG_DRIVE_STRENGTH	Pin config drive strength
PINCTRL_CONFIG_VOLTAGE_STATUS	Pin config voltage status

Enumeration pm_ioctl_id

Table 67: Enumeration pm_ioctl_id Values

Value	Description
IOCTL_GET_RPU_OPER_MODE	Get RPU mode
IOCTL_SET_RPU_OPER_MODE	Set RPU mode
IOCTL_RPU_BOOT_ADDR_CONFIG	RPU boot address config
IOCTL_TCM_COMB_CONFIG	TCM config
IOCTL_SET_TAPDELAY_BYPASS	TAP delay bypass
IOCTL_SET_SGMII_MODE	SGMII mode
IOCTL_SD_DLL_RESET	SD DLL reset
IOCTL_SET_SD_TAPDELAY	SD TAP delay
IOCTL_SET_PLL_FRAC_MODE	Set PLL frac mode
IOCTL_GET_PLL_FRAC_MODE	Get PLL frac mode
IOCTL_SET_PLL_FRAC_DATA	Set PLL frac data

Table 67: Enumeration pm_ioctl_id Values (cont'd)

Value	Description
IOCTL_GET_PLL_FRAC_DATA	Get PLL frac data
IOCTL_WRITE_GGS	Write GGS
IOCTL_READ_GGS	Read GGS
IOCTL_WRITE_PGGS	Write PGGS
IOCTL_READ_PGGS	Read PGGS
IOCTL_ULPI_RESET	ULPI reset
IOCTL_SET_BOOT_HEALTH_STATUS	Set boot status
IOCTL_AFI	AFI
IOCTL_PROBE_COUNTER_READ	Probe counter read
IOCTL_PROBE_COUNTER_WRITE	Probe counter write
IOCTL_OSPI_MUX_SELECT	OSPI mux select
IOCTL_USB_SET_STATE	USB set state
IOCTL_GET_LAST_RESET_REASON	Get last reset reason
IOCTL_AIE_ISR_CLEAR	AIE ISR clear
IOCTL_REGISTER_SGI	Register SGI to ATF
IOCTL_SET_FEATURE_CONFIG	Set runtime feature config
IOCTL_GET_FEATURE_CONFIG	Get runtime feature config
IOCTL_GET_RPU_OPER_MODE	Get RPU mode
IOCTL_SET_RPU_OPER_MODE	Set RPU mode
IOCTL_RPU_BOOT_ADDR_CONFIG	RPU boot address config
IOCTL_TCM_COMB_CONFIG	TCM config
IOCTL_SET_TAPDELAY_BYPASS	TAP delay bypass
IOCTL_SET_SGMII_MODE	SGMII mode
IOCTL_SD_DLL_RESET	SD DLL reset
IOCTL_SET_SD_TAPDELAY	SD TAP delay
IOCTL_SET_PLL_FRAC_MODE	Set PLL frac mode
IOCTL_GET_PLL_FRAC_MODE	Get PLL frac mode
IOCTL_SET_PLL_FRAC_DATA	Set PLL frac data
IOCTL_GET_PLL_FRAC_DATA	Get PLL frac data
IOCTL_WRITE_GGS	Write GGS
IOCTL_READ_GGS	Read GGS
IOCTL_WRITE_PGGS	Write PGGS
IOCTL_READ_PGGS	Read PGGS
IOCTL_ULPI_RESET	ULPI reset
IOCTL_SET_BOOT_HEALTH_STATUS	Set boot status
IOCTL_AFI	AFI
IOCTL_PROBE_COUNTER_READ	Probe counter read
IOCTL_PROBE_COUNTER_WRITE	Probe counter write
IOCTL_OSPI_MUX_SELECT	OSPI mux select

Table 67: Enumeration pm_ioctl_id Values (cont'd)

Value	Description
IOCTL_USB_SET_STATE	USB set state
IOCTL_GET_LAST_RESET_REASON	Get last reset reason
IOCTL_AIE_ISR_CLEAR	AIE ISR clear
IOCTL_REGISTER_SGI	Register SGI to ATF
IOCTL_SET_FEATURE_CONFIG	Set feature config
IOCTL_GET_FEATURE_CONFIG	Get feature config

Enumeration pm_feature_id

Table 68: Enumeration pm_feature_id Values

Value	Description
XPM_FEATURE_INVALID	Invalid ID
XPM_FEATURE_OVERTEMP_STATUS	Over temperature status
XPM_FEATURE_OVERTEMP_VALUE	Over temperature limit
XPM_FEATURE_EXTWDT_STATUS	External watchdog status
XPM_FEATURE_EXTWDT_VALUE	External watchdog interval

Definitions

Define XST_PM_INTERNAL

Definition

```
#define XST_PM_INTERNAL 2000L
```

Description

Define XST_PM_CONFLICT

Definition

```
#define XST_PM_CONFLICT 2001L
```

Description

Define XST_PM_NO_ACCESS

Definition

```
#define XST_PM_NO_ACCESS 2002L
```

Description

Define XST_PM_INVALID_NODE

Definition

```
#define XST_PM_INVALID_NODE 2003L
```

Description

Define XST_PM_DOUBLE_REQ

Definition

```
#define XST_PM_DOUBLE_REQ 2004L
```

Description

Define XST_PM_ABORT_SUSPEND

Definition

```
#define XST_PM_ABORT_SUSPEND 2005L
```

Description

Define XST_PM_TIMEOUT

Definition

```
#define XST_PM_TIMEOUT 2006L
```

Description

Define XST_PM_NODE_USED

Definition

```
#define XST_PM_NODE_USED 2007L
```

Description

XiLPM Versal ACAP APIs

Embedded Energy

Management Interface EEMI API Reference Guide

Table 69: Quick Function Reference

Type	Name	Arguments
XStatus	XPm_InitXilpm	XIpiPsu * IpiInst
enum	XPm_GetBootStatus	void
XStatus	XPm_GetChipID	u32 * IDCode u32 * Version
XStatus	XPm_GetApiVersion	u32 * Version
XStatus	XPm_RequestNode	const u32 DeviceId const u32 Capabilities const u32 QoS const u32 Ack
XStatus	XPm_ReleaseNode	const u32 DeviceId
XStatus	XPm_SetRequirement	const u32 DeviceId const u32 Capabilities const u32 QoS const u32 Ack
XStatus	XPm_GetNodeStatus	const u32 DeviceId XPm_NodeStatus *const NodeStatus

Table 69: Quick Function Reference (cont'd)

Type	Name	Arguments
XStatus	XPm_ResetAssert	const u32 ResetId const u32 Action
XStatus	XPm_ResetGetStatus	const u32 ResetId u32 *const State
XStatus	XPm_PinCtrlRequest	const u32 PinId
XStatus	XPm_PinCtrlRelease	const u32 PinId
XStatus	XPm_PinCtrlSetFunction	const u32 PinId const u32 FunctionId
XStatus	XPm_PinCtrlGetFunction	const u32 PinId u32 *const FunctionId
XStatus	XPm_PinCtrlSetParameter	const u32 PinId const u32 ParamId const u32 ParamVal
XStatus	XPm_PinCtrlGetParameter	const u32 PinId const u32 ParamId u32 *const ParamVal
XStatus	XPm_DeVioctl	const u32 DeviceId const pm_ioctl_id IoctlId const u32 Arg1 const u32 Arg2 u32 *const Response
XStatus	XPm_ClockEnable	const u32 ClockId
XStatus	XPm_ClockDisable	const u32 ClockId
XStatus	XPm_ClockGetStatus	const u32 ClockId u32 *const State
XStatus	XPm_ClockSetDivider	const u32 ClockId const u32 Divider

Table 69: Quick Function Reference (cont'd)

Type	Name	Arguments
XStatus	XPm_ClockGetDivider	const u32 ClockId u32 *const Divider
XStatus	XPm_ClockSetParent	const u32 ClockId const u32 ParentIdx
XStatus	XPm_ClockGetParent	const u32 ClockId u32 *const ParentIdx
XStatus	XPm_ClockGetRate	const u32 ClockId u32 *const Rate
XStatus	XPm_ClockSetRate	const u32 ClockId const u32 Rate
XStatus	XPm_PllSetParameter	const u32 ClockId const enum XPm_PllConfigParams ParamId const u32 Value
XStatus	XPm_PllGetParameter	const u32 ClockId const enum XPm_PllConfigParams ParamId u32 *const Value
XStatus	XPm_PllSetMode	const u32 ClockId const u32 Value
XStatus	XPm_PllGetMode	const u32 ClockId u32 *const Value
XStatus	XPm_SelfSuspend	const u32 DeviceId const u32 Latency const u8 State const u64 Address
XStatus	XPm_RequestWakeUp	const u32 TargetDevId const u8 SetAddress const u64 Address const u32 Ack
void	XPm_SuspendFinalize	void

Table 69: Quick Function Reference (cont'd)

Type	Name	Arguments
XStatus	XPm_RequestSuspend	const u32 TargetSubsystemId const u32 Ack const u32 Latency const u32 State
XStatus	XPm_AbortSuspend	const enum XPmAbortReason Reason
XStatus	XPm_ForcePowerDown	const u32 TargetDevId const u32 Ack
XStatus	XPm_SystemShutdown	const u32 Type const u32 SubType
XStatus	XPm_SetWakeUpSource	const u32 TargetDeviceId const u32 DeviceId const u32 Enable
XStatus	XPm_Query	const u32 QueryId const u32 Arg1 const u32 Arg2 const u32 Arg3 u32 *const Data
XStatus	XPm_SetMaxLatency	const u32 DeviceId const u32 Latency
XStatus	XPm_GetOpCharacteristic	const u32 DeviceId const enum XPmOpCharType Type u32 *const Result
XStatus	XPm_InitFinalize	void
XStatus	XPm_RegisterNotifier	XPmNotifier *const Notifier
XStatus	XPm_UnregisterNotifier	XPmNotifier *const Notifier
void	XPm_InitSuspendCb	const enum XPmSuspendReason Reason const u32 Latency const u32 State const u32 Timeout

Table 69: Quick Function Reference (cont'd)

Type	Name	Arguments
void	XPm_AcknowledgeCb	const u32 Node const XStatus Status const u32 Oppoint
void	XPm_NotifyCb	const u32 Node const u32 Event const u32 Oppoint
XStatus	XPm_FeatureCheck	const u32 FeatureId u32 * Version

Functions

XPm_InitXilpm

Prototype

```
XStatus XPm_InitXilpm(Xilpm *XilpmInst);
```

Parameters

XPm_InitXilpm

Prototype

```
enum
    XPrnBootStatus
XPm_GetBootStatus(void);
```

Returns

XPm_GetChipID

Prototype

```
XStatus XPm_GetChipID(u32 *IDCode, u32 *Version);
```

Parameters

XPm_GetChipID

Table 71: XPm_GetChipID Arguments

Type	Name	Description
u32 *	IDCode	Returns the chip ID code.
u32 *	Version	Returns the chip version.

Returns

XPm_GetApiVersion

Prototype

```
XStatus XPm_GetApiVersion(u32 *Version);
```

Parameters

XPm_GetApiVersion

Table 72: XPm_GetApiVersion Arguments

Type	Name	Description
u32 *	Version	Returns the API 32-bit version number.

Returns

XPm_RequestNode

Prototype

```
XStatus XPm_RequestNode(const u32 DeviceId, const u32 Capabilities, const u32 QoS, const u32 Ack);
```

Parameters

XPm_RequestNode

Table 73: XPm_RequestNode Arguments

Type	Name	Description
const u32	DeviceId	Device which needs to be requested
const u32	Capabilities	Device Capabilities, can be combined <ul style="list-style-type: none"> PM_CAP_ACCESS : full access / functionality PM_CAP_CONTEXT : preserve context PM_CAP_WAKEUP : emit wake interrupts
const u32	QoS	Quality of Service (0-100) required
const u32	Ack	Requested acknowledge type

Returns

XPm_ReleaseNode

Prototype

```
XStatus XPm_ReleaseNode(const u32 DeviceId);
```

Parameters

XPm_ReleaseNode

Table 74: XPm_ReleaseNode Arguments

Type	Name	Description
const u32	DeviceId	Device which needs to be released

Returns

XPm_SetRequirement

Prototype

```
XStatus XPm_SetRequirement(const u32 DeviceId, const u32 Capabilities,
const u32 QoS, const u32 Ack);
```

Parameters

XPm_SetRequirement

Table 75: XPm_SetRequirement Arguments

Type	Name	Description
const u32	DeviceId	Device for which requirement needs to be set
const u32	Capabilities	Device Capabilities, can be combined <ul style="list-style-type: none"> PM_CAP_ACCESS : full access / functionality PM_CAP_CONTEXT : preserve context PM_CAP_WAKEUP : emit wake interrupts
const u32	QoS	Quality of Service (0-100) required
const u32	Ack	Requested acknowledge type

Returns

XPm_GetNodeStatus

Prototype

```
XStatus XPm_GetNodeStatus(const u32 DeviceId, XPm_NodeStatus *const
NodeStatus);
```

Parameters

XPm_GetNodeStatus

Table 76: XPm_GetNodeStatus Arguments

Type	Name	Description
const u32	DeviceId	Device for which status is requested
XPm_NodeStatus *const	NodeStatus	<p>Structure pointer to store device status</p> <ul style="list-style-type: none"> Status - The current power state of the device <ul style="list-style-type: none"> For CPU nodes: <ul style="list-style-type: none"> 0 : if CPU is powered down, 1 : if CPU is active (powered up), 8 : if CPU is suspending (powered up) For power islands and power domains: <ul style="list-style-type: none"> 0 : if island is powered down, 2 : if island is powered up For slaves: <ul style="list-style-type: none"> 0 : if slave is powered down, 1 : if slave is powered up, 9 : if slave is in retention Requirement - Requirements placed on the device by the caller Usage <ul style="list-style-type: none"> 0 : node is not used by any PU, 1 : node is used by caller exclusively, 2 : node is used by other PU(s) only, 3 : node is used by caller and by other PU(s)

Returns

XPm_ResetAssert

Prototype

```
XStatus XPm_ResetAssert(const u32 ResetId, const u32 Action);
```

Parameters

XPm_ResetAssert

Table 77: XPm_ResetAssert Arguments

Type	Name	Description
const u32	ResetId	Reset ID
const u32	Action	Reset action to be taken <ul style="list-style-type: none"> PM_RESET_ACTION_RELEASE for Release Reset PM_RESET_ACTION_ASSERT for Assert Reset PM_RESET_ACTION_PULSE for Pulse Reset

Returns

XPm_ResetGetStatus

Prototype

```
XStatus XPm_ResetGetStatus(const u32 ResetId, u32 *const State);
```

Parameters

XPm_ResetGetStatus

Table 78: XPm_ResetGetStatus Arguments

Type	Name	Description
const u32	ResetId	Reset ID

Table 78: **XPm_ResetGetStatus Arguments** (cont'd)

Type	Name	Description
u32 *const	State	Pointer to store the status of specified reset <ul style="list-style-type: none"> 0 for reset released 1 for reset asserted

Returns

XPm_PinCtrlRequest

Prototype

```
XStatus XPm_PinCtrlRequest(const u32 PinId);
```

Parameters

XPm_PinCtrlRequest

Table 79: **XPm_PinCtrlRequest Arguments**

Type	Name	Description
const u32	PinId	Pin ID

Returns

XPm_PinCtrlRelease

Prototype

```
XStatus XPm_PinCtrlRelease(const u32 PinId);
```

Parameters

XPm_PinCtrlRelease

Table 80: XPm_PinCtrlRelease Arguments

Type	Name	Description
const u32	PinId	Pin ID

Returns

XPm_PinCtrlSetFunction

Prototype

```
XStatus XPm_PinCtrlSetFunction(const u32 PinId, const u32 FunctionId);
```

Parameters

XPm_PinCtrlSetFunction

Table 81: XPm_PinCtrlSetFunction Arguments

Type	Name	Description
const u32	PinId	Pin ID
const u32	FunctionId	Function ID which needs to be set

Returns

XPm_PinCtrlGetFunction

Prototype

```
XStatus XPm_PinCtrlGetFunction(const u32 PinId, u32 *const FunctionId);
```

Parameters

XPm_PinCtrlGetFunction

Table 82: XpM_PinCtrlGetFunction Arguments

Type	Name	Description
const u32	PinId	Pin ID
u32 *const	FunctionId	Pointer to Function ID

Returns

XPm_PinCtrlSetParameter

ParamId	ParamVal
PINCTRL_CONFIG_SLEW_RATE	PINCTRL_SLEW_RATE_SLOW, PINCTRL_SLEW_RATE_FAST
PINCTRL_CONFIG_BIAS_STATUS	PINCTRL_BIAS_DISABLE, PINCTRL_BIAS_ENABLE
PINCTRL_CONFIG_PULL_CTRL	PINCTRL_BIAS_PULL_DOWN, PINCTRL_BIAS_PULL_UP
PINCTRL_CONFIG_SCHMITT_CMOS	PINCTRL_INPUT_TYPE_CMOS, PINCTRL_INPUT_TYPE_SCHMITT
PINCTRL_CONFIG_DRIVE_STRENGTH	PINCTRL_DRIVE_STRENGTH_TRISTATE, PINCTRL_DRIVE_STRENGTH_4MA, PINCTRL_DRIVE_STRENGTH_8MA, PINCTRL_DRIVE_STRENGTH_12MA
PINCTRL_CONFIG_TRI_STATE	PINCTRL_TRI_STATE_DISABLE, PINCTRL_TRI_STATE_ENABLE

Prototype

```
XStatus XPm_PinCtrlSetParameter(const u32 PinId, const u32 ParamId, const u32 ParamVal);
```

Parameters

XPm_PinCtrlSetParameter

Table 83: XPm_PinCtrlSetParameter Arguments

Type	Name	Description
const u32	PinId	Pin ID
const u32	ParamId	Parameter ID
const u32	ParamVal	Value of the parameter

Returns

XPm_PinCtrlGetParameter

ParamId	ParamVal
PINCTRL_CONFIG_SLEW_RATE	PINCTRL_SLEW_RATE_SLOW, PINCTRL_SLEW_RATE_FAST
PINCTRL_CONFIG_BIAS_STATUS	PINCTRL_BIAS_DISABLE, PINCTRL_BIAS_ENABLE
PINCTRL_CONFIG_PULL_CTRL	PINCTRL_BIAS_PULL_DOWN, PINCTRL_BIAS_PULL_UP
PINCTRL_CONFIG_SCHMITT_CMOS	PINCTRL_INPUT_TYPE_CMOS, PINCTRL_INPUT_TYPE_SCHMITT
PINCTRL_CONFIG_DRIVE_STRENGTH	PINCTRL_DRIVE_STRENGTH_TRISTATE, PINCTRL_DRIVE_STRENGTH_4MA, PINCTRL_DRIVE_STRENGTH_8MA, PINCTRL_DRIVE_STRENGTH_12MA
PINCTRL_CONFIG_VOLTAGE_STATUS	1 for 1.8v mode, 0 for 3.3v mode
PINCTRL_CONFIG_TRI_STATE	PINCTRL_TRI_STATE_DISABLE, PINCTRL_TRI_STATE_ENABLE

Prototype

```
XStatus XPm_PinCtrlGetParameter(const u32 PinId, const u32 ParamId, u32 *const ParamVal);
```

Parameters

XPm_PinCtrlGetParameter

Table 84: XPm_PinCtrlGetParameter Arguments

Type	Name	Description
const u32	PinId	Pin ID
const u32	ParamId	Parameter ID
u32 *const	ParamVal	Pointer to the value of the parameter

Returns

XPm_DevIoctl

Prototype

```
XStatus XPm_DevIoctl(const u32 DeviceId, const pm_ioctl_id IoctlId, const u32 Arg1, const u32 Arg2, u32 *const Response);
```

Parameters

XPm_Development

Table 85: XPm_Development Arguments

Type	Name	Description
const u32	DeviceId	ID of the device
const pm_ioctl_id	IoctlId	IOCTL function ID
const u32	Arg1	Argument 1
const u32	Arg2	Argument 2
u32 *const	Response	Ioctl response

Returns

XPm_ClockEnable

Prototype

```
XStatus XPm_ClockEnable(const u32 ClockId);
```

Parameters

XPm_ClockEnable

Table 86: XPm_ClockEnable Arguments

Type	Name	Description
const u32	ClockId	Clock ID

Returns

XPm_ClockDisable

Prototype

```
XStatus XPm_ClockDisable(const u32 ClockId);
```

Parameters

XPm_Cl ockDi sabl e

Table 87: XPm_ClockDisable Arguments

Type	Name	Description
const u32	ClockId	Clock ID

Returns

XPm_ClockGetStatus

Prototype

```
XStatus XPm_Cl ockGetStatus(const u32 Cl ockI d, u32 *const State);
```

Parameters

XPm_Cl ockGet Status

Table 88: XPm_ClockGetStatus Arguments

Type	Name	Description
const u32	ClockId	Clock ID
u32 *const	State	Pointer to store the clock state <ul style="list-style-type: none"> 1 for enable and 0 for disable

Returns

XPm_ClockSetDivider

Prototype

```
XStatus XPm_Cl ockSetDi vi der(const u32 Cl ockI d, const u32 Di vi der);
```


Parameters

XPm_ClockSetDivider

Table 89: XPm_ClockSetDivider Arguments

Type	Name	Description
const u32	ClockId	Clock ID
const u32	Divider	Value of the divider

Returns

XPm_ClockGetDivider

Prototype

```
XStatus XPm_ClockGetDivider(const u32 ClockId, u32 *const Divider);
```

Parameters

XPm_ClockGetDivider

Table 90: XPm_ClockGetDivider Arguments

Type	Name	Description
const u32	ClockId	Clock ID
u32 *const	Divider	Pointer to store divider value

Returns

XPm_ClockSetParent

Prototype

```
XStatus XPm_ClockSetParent(const u32 ClockId, const u32 ParentIdx);
```

Parameters

XPm_ClockSetParent

Table 91: XPm_ClockSetParent Arguments

Type	Name	Description
const u32	ClockId	Clock ID
const u32	ParentIdx	Parent clock index

Returns

XPm_ClockGetParent

Prototype

```
XStatus XPm_ClockGetParent(const u32 ClockId, u32 *const ParentIdx);
```

Parameters

XPm_ClockGetParent

Table 92: XPm_ClockGetParent Arguments

Type	Name	Description
const u32	ClockId	Clock ID
u32 *const	ParentIdx	Pointer to store the parent clock index

Returns

XPm_ClockGetRate

Prototype

```
XStatus XPm_ClockGetRate(const u32 ClockId, u32 *const Rate);
```

Parameters

XPm_ClockGetRate

Table 93: XPm_ClockGetRate Arguments

Type	Name	Description
const u32	ClockId	Clock ID
u32 *const	Rate	Pointer to store the rate clock

Returns

XPm_ClockSetRate

Prototype

```
XStatus XPm_ClockSetRate(const u32 ClockId, const u32 Rate);
```

Parameters

XPm_ClockSetRate

Table 94: XPm_ClockSetRate Arguments

Type	Name	Description
const u32	ClockId	Clock ID
const u32	Rate	Clock rate

Returns

XPm_PllSetParameter

Prototype

```
XStatus XPm_PllSetParameter(const u32 ClockId, const enum
XPm_PllConfigParams ParamId, const u32 Value);
```

Parameters

XPm_PII SetParameter

Table 95: XPm_PIISetParameter Arguments

Type	Name	Description
const u32	ClockId	Clock ID
const enum XPm_PII ConfigParams	ParamId	Parameter ID <ul style="list-style-type: none"> PM_PLL_PARAM_ID_DIV2 PM_PLL_PARAM_ID_FBDIV PM_PLL_PARAM_ID_DATA PM_PLL_PARAM_ID_PRE_SRC PM_PLL_PARAM_ID_POST_SRC PM_PLL_PARAM_ID_LOCK_DLY PM_PLL_PARAM_ID_LOCK_CNT PM_PLL_PARAM_ID_LFHF PM_PLL_PARAM_ID_CP PM_PLL_PARAM_ID_RES
const u32	Value	Value of parameter (See register description for possible values)

Returns

XPm_PIIGetParameter

Prototype

```
XStatus XPm_PIIGetParameter(const u32 ClockId, const enum
XPm_PII ConfigParams ParamId, u32 *const Value);
```

Parameters

XPm_PII GetParameter

Table 96: XPm_PIIGetParameter Arguments

Type	Name	Description
const u32	ClockId	Clock ID

Table 96: XPm_PllGetParameter Arguments (cont'd)

Type	Name	Description
const enum XPm_PllConfigParams	ParamId	Parameter ID <ul style="list-style-type: none"> PM_PLL_PARAM_ID_DIV2 PM_PLL_PARAM_ID_FBDIV PM_PLL_PARAM_ID_DATA PM_PLL_PARAM_ID_PRE_SRC PM_PLL_PARAM_ID_POST_SRC PM_PLL_PARAM_ID_LOCK_DLY PM_PLL_PARAM_ID_LOCK_CNT PM_PLL_PARAM_ID_LFHF PM_PLL_PARAM_ID_CP PM_PLL_PARAM_ID_RES
u32 *const	Value	Pointer to store parameter value (See register description for possible values)

Returns

XPm_PllSetMode

Prototype

```
XStatus XPm_PllSetMode(const u32 ClockId, const u32 Value);
```

Parameters

XPm_PllSetMode

Table 97: XPm_PllSetMode Arguments

Type	Name	Description
const u32	ClockId	Clock ID
const u32	Value	Mode which need to be set <ul style="list-style-type: none"> 0 for Reset mode 1 for Integer mode 2 for Fractional mode

Returns

XPm_PllGetMode

Prototype

```
XStatus XPm_PllGetMode(const u32 ClockId, u32 *const Value);
```

Parameters

XPm_PllGetMode

Table 98: XPm_PllGetMode Arguments

Type	Name	Description
const u32	ClockId	Clock ID
u32 *const	Value	Pointer to store the value of mode <ul style="list-style-type: none"> 0 for Reset mode 1 for Integer mode 2 for Fractional mode

Returns

XPm_SelfSuspend

Prototype

```
XStatus XPm_SelfSuspend(const u32 DeviceId, const u32 Latency, const u8 State, const u64 Address);
```

Parameters

XPm_SelfSuspend

Table 99: XPm_SelfSuspend Arguments

Type	Name	Description
const u32	DeviceId	Device ID of the CPU
const u32	Latency	Maximum wake-up latency requirement in us(microsecs)
const u8	State	Instead of specifying a maximum latency, a CPU can also explicitly request a certain power state.
const u64	Address	Address from which to resume when woken up.

Returns

XPm_RequestWakeUp

Prototype

```
XStatus XPm_RequestWakeUp(const u32 TargetDevId, const u8 SetAddress, const u64 Address, const u32 Ack);
```

Parameters

XPm_RequestWakeUp

Table 100: XPm_RequestWakeUp Arguments

Type	Name	Description
const u32	TargetDevId	Device ID of the CPU or PU to be powered/woken up.
const u8	SetAddress	Specifies whether the start address argument is being passed. <ul style="list-style-type: none"> 0 : do not set start address 1 : set start address
const u64	Address	Address from which to resume when woken up. Will only be used if set_address is 1.
const u32	Ack	Requested acknowledge type

Returns

XPm_SuspendFinalize

Note

Prototype

```
void XPm_SuspendFinalize(void);
```

Returns

XPm_RequestSuspend

Prototype

```
XStatus XPm_RequestSuspend(const u32 TargetSubsystemId, const u32 Ack,
const u32 Latency, const u32 State);
```

Parameters

XPm_RequestSuspend

Table 101: XPm_RequestSuspend Arguments

Type	Name	Description
const u32	TargetSubsystemId	Subsystem ID of the target
const u32	Ack	Requested acknowledge type
const u32	Latency	Maximum wake-up latency requirement in us(microsecs)
const u32	State	Power State

Returns

XPm_AbortSuspend

Prototype

```
XStatus XPm_AbortSuspend(const enum XPm_AbortReason Reason);
```

Parameters

XPm_AbortSuspend

Table 102: XPm_AbortSuspend Arguments

Type	Name	Description
const enum XPm_AbortReason	Reason	Reason code why the suspend can not be performed or completed <ul style="list-style-type: none"> ABORT_REASON_WKUP_EVENT : local wakeup-event received ABORT_REASON_PU_BUSY : PU is busy ABORT_REASON_NO_PWRDN : no external powerdown supported ABORT_REASON_UNKNOWN : unknown error during suspend procedure

Returns

XPm_ForcePowerDown

Note

Prototype

```
XStatus XPm_ForcePowerDown(const u32 TargetDevId, const u32 Ack);
```

Parameters

XPm_ForcePowerDown

Table 103: XPm_ForcePowerDown Arguments

Type	Name	Description
const u32	TargetDevId	Device ID of the PU node to be forced powered down.
const u32	Ack	Requested acknowledge type

Returns

XPm_SystemShutdown

Prototype

```
XStatus XPm_SystemShutdown(const u32 Type, const u32 SubType);
```

Parameters

XPm_SystemShutdown

Table 104: XPm_SystemShutdown Arguments

Type	Name	Description
const u32	Type	Shutdown type (shutdown/restart)
const u32	SubType	Shutdown subtype (subsystem-only/PU-only/system)

Returns

XPm_SetWakeUpSource

Prototype

```
XStatus XPm_SetWakeUpSource(const u32 TargetDeviceId, const u32 DeviceId, const u32 Enable);
```

Parameters

XPm_SetWakeUpSource

Table 105: XPm_SetWakeUpSource Arguments

Type	Name	Description
const u32	TargetDeviceId	Device ID of the target
const u32	DeviceId	Device ID used as wakeup source
const u32	Enable	1 - Enable, 0 - Disable

Returns

XPm_Query

Prototype

```
XStatus XPm_Query(const u32 QueryId, const u32 Arg1, const u32 Arg2, const u32 Arg3, u32 *const Data);
```

Parameters

XPm_Query

Table 106: XPm_Query Arguments

Type	Name	Description
const u32	QueryId	The type of data to query
const u32	Arg1	Query argument 1
const u32	Arg2	Query argument 2
const u32	Arg3	Query argument 3
u32 *const	Data	Pointer to the output data

Returns

XPm_SetMaxLatency

Note

Prototype

```
XStatus XPm_SetMaxLatency(const u32 DeviceId, const u32 Latency);
```

Parameters

XPm_SetMaxLatency

Table 107: XPm_SetMaxLatency Arguments

Type	Name	Description
const u32	DeviceId	Device ID.
const u32	Latency	Maximum wake-up latency required.

Returns

XPm_GetOpCharacteristic

Note

Prototype

```
XStatus XPm_GetOpCharacteristic(const u32 DeviceId, const enum
XPmOpCharType Type, u32 *const Result);
```

Parameters

XPm_GetOpCharacteristic

Table 108: XPm_GetOpCharacteristic Arguments

Type	Name	Description
const u32	DeviceId	Device ID.
const enum XPmOpCharType	Type	Type of operating characteristic requested: <ul style="list-style-type: none"> power (current power consumption), latency (current latency in micro seconds to return to active state), temperature (current temperature in Celsius (Q8.7 format)),
u32 *const	Result	Used to return the requested operating characteristic.

Returns

XPm_InitFinalize

Prototype

```
XStatus XPm_InitFinalize(void);
```

Returns

XPm_RegisterNotifier

Note

Prototype

```
XStatus XPm_RegisterNotifier(XPm_Notifier *const Notifier);
```

Parameters

XPm_RegisterNotifier

Table 109: XPm_RegisterNotifier Arguments

Type	Name	Description
XPm_Notifier *const	Notifier	Pointer to the notifier object to be associated with the requested notification.

Returns

XPm_UnregisterNotifier

Prototype

```
XStatus XPm_UnregisterNotifier(XPm_Notifier *const Notifier);
```

Parameters

XPm_UnregisterNotifier

Table 110: XPm_UnregisterNotifier Arguments

Type	Name	Description
XPm_Notifier *const	Notifier	Pointer to the notifier object associated with the previously requested notification

Returns

XPm_InitSuspendCb

Note

Prototype

```
void XPm_InitSuspendCb(const enum XPmSuspendReason Reason, const u32 Latency, const u32 State, const u32 Timeout);
```

Parameters

XPm_InitSuspendCb

Table 111: XPm_InitSuspendCb Arguments

Type	Name	Description
const enum XPmSuspendReason	Reason	Suspend reason: <ul style="list-style-type: none"> SUSPEND_REASON_PU_REQ : Request by another PU SUSPEND_REASON_ALERT : Unrecoverable SysMon alert SUSPEND_REASON_SHUTDOWN : System shutdown SUSPEND_REASON_RESTART : System restart
const u32	Latency	Maximum wake-up latency in us(micro secs). This information can be used by the PU to decide what level of context saving may be required.

Table 111: XPm_InitSuspendCb Arguments (cont'd)

Type	Name	Description
const u32	State	Targeted sleep/suspend state.
const u32	Timeout	Timeout in ms, specifying how much time a PU has to initiate its suspend procedure before it's being considered unresponsive.

Returns

Prototype

```
void XPm_NotifyCb(const u32 Node, const u32 Event, const u32 Oppoint);
```

Parameters

XPm_NotifyCb

Table 113: XPm_NotifyCb Arguments

Type	Name	Description
const u32	Node	ID of the device the event notification is related to.
const u32	Event	ID of the event
const u32	Oppoint	Current operating state of the device.

Returns

XPm_FeatureCheck

Prototype

```
XStatus XPm_FeatureCheck(const u32 FeatureId, u32 *Version);
```

Parameters

XPm_FeatureCheck

Table 114: XPm_FeatureCheck Arguments

Type	Name	Description
const u32	FeatureId	The feature ID (API-ID)
u32 *	Version	Pointer to the output data where version of feature store. For the supported feature get non zero value in version, But if version is 0U that means feature not supported.

Returns

Enumerations

Enumeration XPmAbortReason

Table 115: Enumeration XPmAbortReason Values

Value	Description
ABORT_REASON_WKUP_EVENT	Wakeup Event
ABORT_REASON_PU_BUSY	Processor Busy
ABORT_REASON_NO_PWRDN	No Powerdown
ABORT_REASON_UNKNOWN	Unknown Reason

Enumeration XPmBootStatus

Table 116: Enumeration XPmBootStatus Values

Value	Description
PM_INITIAL_BOOT	Boot is a fresh system startup
PM_RESUME	Boot is a resume
PM_BOOT_ERROR	Error. Cause cannot be identified

Enumeration XPmRequestAck

Table 117: Enumeration XPmRequestAck Values

Value	Description
REQUEST_ACK_NO	No Ack
REQUEST_ACK_BLOCKING	Blocking Ack
REQUEST_ACK_NON_BLOCKING	Non blocking Ack
REQUEST_ACK_CB_CERROR	Callback Error

Enumeration XPmCapability

Table 118: Enumeration XPmCapability Values

Value	Description
PM_CAP_ACCESS	Full access
PM_CAP_CONTEXT	Configuration and contents retained
PM_CAP_WAKEUP	Enabled as a wake-up source
PM_CAP_UNUSABLE	Not usable
PM_CAP_SECURE	Secure access type (non-secure/secure)
PM_CAP_COHERENT	Device Coherency
PM_CAP_VIRTUALIZED	Device Virtualization

Enumeration XPmDeviceUsage

Table 119: Enumeration XPmDeviceUsage Values

Value	Description
PM_USAGE_CURRENT_SUBSYSTEM	Current subsystem is using
PM_USAGE_OTHER_SUBSYSTEM	Other subsystem is using

Enumeration XPmResetActions

Table 120: Enumeration XPmResetActions Values

Value	Description
PM_RESET_ACTION_RELEASE	Reset action release
PM_RESET_ACTION_ASSERT	Reset action assert
PM_RESET_ACTION_PULSE	Reset action pulse

Enumeration XPmSuspendReason

Table 121: Enumeration XPmSuspendReason Values

Value	Description
SUSPEND_REASON_PU_REQ	Processor request
SUSPEND_REASON_ALERT	Alert
SUSPEND_REASON_SYS_SHUTDOWN	System shutdown

Enumeration XPmApiCbId_t

Table 122: Enumeration XPmApiCbId_t Values

Value	Description
PM_INIT_SUSPEND_CB	Suspend callback
PM_ACKNOWLEDGE_CB	Acknowledge callback
PM_NOTIFY_CB	Notify callback

Enumeration pm_query_id

Table 123: Enumeration pm_query_id Values

Value	Description
XPM_QID_INVALID	Invalid Query ID
XPM_QID_CLOCK_GET_NAME	Get clock name
XPM_QID_CLOCK_GET_TOPOLOGY	Get clock topology
XPM_QID_CLOCK_GET_FIXEDFACTOR_PARAMS	Get clock fixedfactor parameter
XPM_QID_CLOCK_GET_MUXSOURCES	Get clock mux sources
XPM_QID_CLOCK_GET_ATTRIBUTES	Get clock attributes
XPM_QID_PINCTRL_GET_NUM_PINS	Get total pins
XPM_QID_PINCTRL_GET_NUM_FUNCTIONS	Get total pin functions
XPM_QID_PINCTRL_GET_NUM_FUNCTION_GROUPS	Get total pin function groups
XPM_QID_PINCTRL_GET_FUNCTION_NAME	Get pin function name
XPM_QID_PINCTRL_GET_FUNCTION_GROUPS	Get pin function groups
XPM_QID_PINCTRL_GET_PIN_GROUPS	Get pin groups
XPM_QID_CLOCK_GET_NUM_CLOCKS	Get number of clocks
XPM_QID_CLOCK_GET_MAX_DIVISOR	Get max clock divisor
XPM_QID_PLD_GET_PARENT	Get PLD parent

Enumeration PmPinFunIds

Table 124: Enumeration PmPinFunIds Values

Value	Description
PIN_FUNC_SPI0	Pin function ID of SPI0
PIN_FUNC_SPI0_SS	Pin function ID of SPI0_SS

Table 124: Enumeration PmPinFunIds Values (cont'd)

Value	Description
PIN_FUNC_SPI1	Pin function ID of SPI1
PIN_FUNC_SPI1_SS	Pin function ID of SPI1_SS
PIN_FUNC_CAN0	Pin function ID of CAN0
PIN_FUNC_CAN1	Pin function ID of CAN1
PIN_FUNC_I2C0	Pin function ID of I2C0
PIN_FUNC_I2C1	Pin function ID of I2C1
PIN_FUNC_I2C_PMC	Pin function ID of I2C_PMC
PIN_FUNC_TTC0_CLK	Pin function ID of TTC0_CLK
PIN_FUNC_TTC0_WAV	Pin function ID of TTC0_WAV
PIN_FUNC_TTC1_CLK	Pin function ID of TTC1_CLK
PIN_FUNC_TTC1_WAV	Pin function ID of TTC1_WAV
PIN_FUNC_TTC2_CLK	Pin function ID of TTC2_CLK
PIN_FUNC_TTC2_WAV	Pin function ID of TTC2_WAV
PIN_FUNC_TTC3_CLK	Pin function ID of TTC3_CLK
PIN_FUNC_TTC3_WAV	Pin function ID of TTC3_WAV
PIN_FUNC_WWDT0	Pin function ID of WWDT0
PIN_FUNC_WWDT1	Pin function ID of WWDT1
PIN_FUNC_SYSMON_I2C0	Pin function ID of SYSMON_I2C0
PIN_FUNC_SYSMON_I2C0_ALERT	Pin function ID of SYSMON_I2C0_AL
PIN_FUNC_UART0	Pin function ID of UART0
PIN_FUNC_UART0_CTRL	Pin function ID of UART0_CTRL
PIN_FUNC_UART1	Pin function ID of UART1
PIN_FUNC_UART1_CTRL	Pin function ID of UART1_CTRL
PIN_FUNC_GPIO0	Pin function ID of GPIO0
PIN_FUNC_GPIO1	Pin function ID of GPIO1
PIN_FUNC_GPIO2	Pin function ID of GPIO2
PIN_FUNC_EMIO0	Pin function ID of EMIO0
PIN_FUNC_GEM0	Pin function ID of GEM0
PIN_FUNC_GEM1	Pin function ID of GEM1
PIN_FUNC_TRACE0	Pin function ID of TRACE0
PIN_FUNC_TRACE0_CLK	Pin function ID of TRACE0_CLK
PIN_FUNC_MDIO0	Pin function ID of MDIO0
PIN_FUNC_MDIO1	Pin function ID of MDIO1
PIN_FUNC_GEM_TSU0	Pin function ID of GEM_TSU0
PIN_FUNC_PCIE0	Pin function ID of PCIE0
PIN_FUNC_SMAP0	Pin function ID of SMAP0
PIN_FUNC_USB0	Pin function ID of USB0
PIN_FUNC_SD0	Pin function ID of SD0
PIN_FUNC_SD0_PC	Pin function ID of SD0_PC

Table 124: Enumeration PmPinFunIds Values (cont'd)

Value	Description
PIN_FUNC_SD0_CD	Pin function ID of SD0_CD
PIN_FUNC_SD0_WP	Pin function ID of SD0_WP
PIN_FUNC_SD1	Pin function ID of SD1
PIN_FUNC_SD1_PC	Pin function ID of SD1_PC
PIN_FUNC_SD1_CD	Pin function ID of SD1_CD
PIN_FUNC_SD1_WP	Pin function ID of SD1_WP
PIN_FUNC_OSPI0	Pin function ID of OSPI0
PIN_FUNC_OSPI0_SS	Pin function ID of OSPI0_SS
PIN_FUNC_QSPI0	Pin function ID of QSPI0
PIN_FUNC_QSPI0_FBCLK	Pin function ID of QSPI0_FBCLK
PIN_FUNC_QSPI0_SS	Pin function ID of QSPI0_SS
PIN_FUNC_TEST_CLK	Pin function ID of TEST_CLK
PIN_FUNC_TEST_SCAN	Pin function ID of TEST_SCAN
PIN_FUNC_TAMPER_TRIGGER	Pin function ID of TAMPER_TRIGGER
MAX_FUNCTION	Max Pin function

Enumeration pm_pinctrl_config_param

Table 125: Enumeration pm_pinctrl_config_param Values

Value	Description
PINCTRL_CONFIG_SLEW_RATE	Pin config slew rate
PINCTRL_CONFIG_BIAS_STATUS	Pin config bias status
PINCTRL_CONFIG_PULL_CTRL	Pin config pull control
PINCTRL_CONFIG_SCHMITT_CMOS	Pin config schmitt CMOS
PINCTRL_CONFIG_DRIVE_STRENGTH	Pin config drive strength
PINCTRL_CONFIG_VOLTAGE_STATUS	Pin config voltage status
PINCTRL_CONFIG_TRI_STATE	Pin config tri state
PINCTRL_CONFIG_MAX	Max Pin config

Enumeration pm_pinctrl_slew_rate

Table 126: Enumeration pm_pinctrl_slew_rate Values

Value	Description
PINCTRL_SLEW_RATE_FAST	Fast slew rate
PINCTRL_SLEW_RATE_SLOW	Slow slew rate

Table 130: Enumeration pm_pinctrl_drive_strength Values

Value	Description
PINCTRL_DRIVE_STRENGTH_TRISTATE	tri-state
PINCTRL_DRIVE_STRENGTH_4MA	4mA
PINCTRL_DRIVE_STRENGTH_8MA	8mA
PINCTRL_DRIVE_STRENGTH_12MA	12mA
PINCTRL_DRIVE_STRENGTH_MAX	Max value

Enumeration pm_pinctrl_tri_state

Table 131: Enumeration pm_pinctrl_tri_state Values

Value	Description
PINCTRL_TRI_STATE_DISABLE	Tri state disable
PINCTRL_TRI_STATE_ENABLE	Tri state enable

Enumeration pm_ioctl_id

Table 132: Enumeration pm_ioctl_id Values

Value	Description
IOCTL_GET_RPU_OPER_MODE	Get RPU mode
IOCTL_SET_RPU_OPER_MODE	Set RPU mode
IOCTL_RPU_BOOT_ADDR_CONFIG	RPU boot address config
IOCTL_TCM_COMB_CONFIG	TCM config
IOCTL_SET_TAPDELAY_BYPASS	TAP delay bypass
IOCTL_SET_SGMII_MODE	SGMII mode
IOCTL_SD_DLL_RESET	SD DLL reset
IOCTL_SET_SD_TAPDELAY	SD TAP delay
IOCTL_SET_PLL_FRAC_MODE	Set PLL frac mode
IOCTL_GET_PLL_FRAC_MODE	Get PLL frac mode
IOCTL_SET_PLL_FRAC_DATA	Set PLL frac data
IOCTL_GET_PLL_FRAC_DATA	Get PLL frac data
IOCTL_WRITE_GGS	Write GGS
IOCTL_READ_GGS	Read GGS
IOCTL_WRITE_PGGS	Write PGGS
IOCTL_READ_PGGS	Read PGGS

Table 132: Enumeration pm_ioctl_id Values (cont'd)

Value	Description
IOCTL_ULPI_RESET	ULPI reset
IOCTL_SET_BOOT_HEALTH_STATUS	Set boot status
IOCTL_AFI	AFI
IOCTL_PROBE_COUNTER_READ	Probe counter read
IOCTL_PROBE_COUNTER_WRITE	Probe counter write
IOCTL_OSPI_MUX_SELECT	OSPI mux select
IOCTL_USB_SET_STATE	USB set state
IOCTL_GET_LAST_RESET_REASON	Get last reset reason
IOCTL_AIE_ISR_CLEAR	AIE ISR clear
IOCTL_REGISTER_SGI	Register SGI to ATF
IOCTL_SET_FEATURE_CONFIG	Set runtime feature config
IOCTL_GET_FEATURE_CONFIG	Get runtime feature config
IOCTL_GET_RPU_OPER_MODE	Get RPU mode
IOCTL_SET_RPU_OPER_MODE	Set RPU mode
IOCTL_RPU_BOOT_ADDR_CONFIG	RPU boot address config
IOCTL_TCM_COMB_CONFIG	TCM config
IOCTL_SET_TAPDELAY_BYPASS	TAP delay bypass
IOCTL_SET_SGMII_MODE	SGMII mode
IOCTL_SD_DLL_RESET	SD DLL reset
IOCTL_SET_SD_TAPDELAY	SD TAP delay
IOCTL_SET_PLL_FRAC_MODE	Set PLL frac mode
IOCTL_GET_PLL_FRAC_MODE	Get PLL frac mode
IOCTL_SET_PLL_FRAC_DATA	Set PLL frac data
IOCTL_GET_PLL_FRAC_DATA	Get PLL frac data
IOCTL_WRITE_GGS	Write GGS
IOCTL_READ_GGS	Read GGS
IOCTL_WRITE_PGGS	Write PGGS
IOCTL_READ_PGGS	Read PGGS
IOCTL_ULPI_RESET	ULPI reset
IOCTL_SET_BOOT_HEALTH_STATUS	Set boot status
IOCTL_AFI	AFI
IOCTL_PROBE_COUNTER_READ	Probe counter read
IOCTL_PROBE_COUNTER_WRITE	Probe counter write
IOCTL_OSPI_MUX_SELECT	OSPI mux select
IOCTL_USB_SET_STATE	USB set state
IOCTL_GET_LAST_RESET_REASON	Get last reset reason
IOCTL_AIE_ISR_CLEAR	AIE ISR clear
IOCTL_REGISTER_SGI	Register SGI to ATF
IOCTL_SET_FEATURE_CONFIG	Set feature config

Table 132: Enumeration pm_ioctl_id Values (cont'd)

Value	Description
IOCTL_GET_FEATURE_CONFIG	Get feature config

Enumeration Xpm_PllConfigParams

Table 133: Enumeration Xpm_PllConfigParams Values

Value	Description
PM_PLL_PARAM_ID_DIV2	PLL param ID DIV2
PM_PLL_PARAM_ID_FBDIV	PLL param ID FBDIV
PM_PLL_PARAM_ID_DATA	PLL param ID DATA
PM_PLL_PARAM_ID_PRE_SRC	PLL param ID PRE_SRC
PM_PLL_PARAM_ID_POST_SRC	PLL param ID POST_SRC
PM_PLL_PARAM_ID_LOCK_DLY	PLL param ID LOCK_DLY
PM_PLL_PARAM_ID_LOCK_CNT	PLL param ID LOCK_CNT
PM_PLL_PARAM_ID_LFHF	PLL param ID LFHF
PM_PLL_PARAM_ID_CP	PLL param ID CP
PM_PLL_PARAM_ID_RES	PLL param ID RES
PM_PLL_PARAM_MAX	PLL param ID max

Enumeration XpmPllMode

Table 134: Enumeration XpmPllMode Values

Value	Description
PM_PLL_MODE_INTEGER	PLL mode integer
PM_PLL_MODE_FRACTIONAL	PLL mode fractional
PM_PLL_MODE_RESET	PLL mode reset

Enumeration XpmInitFunctions

Table 135: Enumeration XpmInitFunctions Values

Value	Description
FUNC_INIT_START	Function ID INIT_START

Table 135: Enumeration XPmInitFunctions Values (cont'd)

Value	Description
FUNC_INIT_FINISH	Function ID INIT_FINISH
FUNC_SCAN_CLEAR	Function ID SCAN_CLEAR
FUNC_BISR	Function ID BISR
FUNC_LBIST	Function ID LBIST
FUNC_MEM_INIT	Function ID MEM_INIT
FUNC_MBIST_CLEAR	Function ID MBIST_CLEAR
FUNC_HOUSECLEAN_PL	Function ID HOUSECLEAN_PL
FUNC_HOUSECLEAN_COMPLETE	Function ID HOUSECLEAN_COMPLETE
FUNC_MAX_COUNT_PMINIT	Function ID MAX

Enumeration XPmOpCharType

Table 136: Enumeration XPmOpCharType Values

Value	Description
PM_OPCHAR_TYPE_POWER	Operating characteristic ID power
PM_OPCHAR_TYPE_TEMP	Operating characteristic ID temperature
PM_OPCHAR_TYPE_LATENCY	Operating characteristic ID latency

Enumeration XPmNotifyEvent

Table 137: Enumeration XPmNotifyEvent Values

Value	Description
EVENT_STATE_CHANGE	State change event
EVENT_ZERO_USERS	Zero user event
EVENT_CPU_IDLE_FORCE_PWRDWN	CPU idle event during force power down

Enumeration XPm_ApiId

Table 138: Enumeration XPm_ApiId Values

Value	Description
PM_API_MIN	0x0

Table 138: Enumeration XPm_ApiId Values (cont'd)

Value	Description
PM_GET_API_VERSION	0x1
PM_SET_CONFIGURATION	0x2
PM_GET_NODE_STATUS	0x3
PM_GET_OP_CHARACTERISTIC	0x4
PM_REGISTER_NOTIFIER	0x5
PM_REQUEST_SUSPEND	0x6
PM_SELF_SUSPEND	0x7
PM_FORCE_POWERDOWN	0x8
PM_ABORT_SUSPEND	0x9
PM_REQUEST_WAKEUP	0xA
PM_SET_WAKEUP_SOURCE	0xB
PM_SYSTEM_SHUTDOWN	0xC
PM_REQUEST_NODE	0xD
PM_RELEASE_NODE	0xE
PM_SET_REQUIREMENT	0xF
PM_SET_MAX_LATENCY	0x10
PM_RESET_ASSERT	0x11
PM_RESET_GET_STATUS	0x12
PM_MMIO_WRITE	0x13
PM_MMIO_READ	0x14
PM_INIT_FINALIZE	0x15
PM_FPGA_LOAD	0x16
PM_FPGA_GET_STATUS	0x17
PM_GET_CHIPID	0x18
PM_SECURE_RSA_AES	0x19
PM_SECURE_SHA	0x1A
PM_SECURE_RSA	0x1B
PM_PINCTRL_REQUEST	0x1C
PM_PINCTRL_RELEASE	0x1D
PM_PINCTRL_GET_FUNCTION	0x1E
PM_PINCTRL_SET_FUNCTION	0x1F
PM_PINCTRL_CONFIG_PARAM_GET	0x20
PM_PINCTRL_CONFIG_PARAM_SET	0x21
PM_IOCTL	0x22
PM_QUERY_DATA	0x23
PM_CLOCK_ENABLE	0x24
PM_CLOCK_DISABLE	0x25
PM_CLOCK_GETSTATE	0x26
PM_CLOCK_SETDIVIDER	0x27

Table 138: Enumeration XPm_ApiId Values (cont'd)

Value	Description
PM_CLOCK_GETDIVIDER	0x28
PM_CLOCK_SETRATE	0x29
PM_CLOCK_GETRATE	0x2A
PM_CLOCK_SETPARENT	0x2B
PM_CLOCK_GETPARENT	0x2C
PM_SECURE_IMAGE	0x2D
PM_FPGA_READ	0x2E
PM_API_RESERVED_1	0x2F
PM_PLL_SET_PARAMETER	0x30
PM_PLL_GET_PARAMETER	0x31
PM_PLL_SET_MODE	0x32
PM_PLL_GET_MODE	0x33
PM_REGISTER_ACCESS	0x34
PM_EFUSE_ACCESS	0x35
PM_ADD_SUBSYSTEM	0x36
PM_DESTROY_SUBSYSTEM	0x37
PM_DESCRIBE_NODES	0x38
PM_ADD_NODE	0x39
PM_ADD_NODE_PARENT	0x3A
PM_ADD_NODE_NAME	0x3B
PM_ADD_REQUIREMENT	0x3C
PM_SET_CURRENT_SUBSYSTEM	0x3D
PM_INIT_NODE	0x3E
PM_FEATURE_CHECK	0x3F
PM_ISO_CONTROL	0x40
PM_ACTIVATE_SUBSYSTEM	0x41
PM_API_MAX	0x42

Definitions

Define PM_VERSION_MAJOR

Definition

```
#define PM_VERSION_MAJOR 1UL
```

Description

Define PM_VERSION_MINOR

Definition

```
#define PM_VERSION_MINOR
```

Description

Define PM_VERSION

Definition

```
#define PM_VERSION((
    PM_VERSION_MAJOR
    << 16) |
    PM_VERSION_MINOR
)
```

Description

Define XPM_MAX_CAPABILITY

Definition

```
#define XPM_MAX_CAPABILITY((u32)
    PM_CAP_ACCESS
    | (u32)
    PM_CAP_CONTEXT
    | (u32)
    PM_CAP_WAKEUP
)
```

Description

Define XPM_MAX_LATENCY

Definition

```
#define XPM_MAX_LATENCY(0xFFFFU)
```

Description

Define XPM_MAX_QOS

Definition

```
#define XPM_MAX_QOS(100U)
```

Description

Define XPM_MIN_CAPABILITY

Definition

```
#define XPM_MIN_CAPABILITY(0U)
```

Description

Define XPM_MIN_LATENCY

Definition

```
#define XPM_MIN_LATENCY(0U)
```

Description

Define XPM_MIN_QOS

Definition

```
#define XPM_MIN_QOS(OU)
```

Description

Define XPM_DEF_CAPABILITY

Definition

```
#define XPM_DEF_CAPABILITY  
XPM_MAX_CAPABILITY
```

Description

Define XPM_DEF_LATENCY

Definition

```
#define XPM_DEF_LATENCY  
XPM_MAX_LATENCY
```

Description

Define XPM_DEF_QOS

Definition

```
#define XPM_DEF_QOS  
XPM_MAX_QOS
```

Description**Define NODE_STATE_OFF****Definition**

```
#define NODE_STATE_OFF(OU)
```

Description**Define NODE_STATE_ON****Definition**

```
#define NODE_STATE_ON(1U)
```

Description**Define PROC_STATE_SLEEP****Definition**

```
#define PROC_STATE_SLEEP  
    NODE_STATE_OFF
```

Description

Define PROC_STATE_ACTIVE

Definition

```
#define PROC_STATE_ACTIVE  
    NODE_STATE_ON
```

Description

Define PROC_STATE_FORCEDOFF

Definition

```
#define PROC_STATE_FORCEDOFF( 7U)
```

Description

Define PROC_STATE_SUSPENDING

Definition

```
#define PROC_STATE_SUSPENDING( 8U)
```

Description

Define PM_SHUTDOWN_TYPE_SHUTDOWN

Definition

```
#define PM_SHUTDOWN_TYPE_SHUTDOWN( 0U)
```

Description

Define PM_SHUTDOWN_TYPE_RESET

Definition

```
#define PM_SHUTDOWN_TYPE_RESET(1U)
```

Description

Define PM_SHUTDOWN_SUBTYPE_RST_SUBSYSTEM

Definition

```
#define PM_SHUTDOWN_SUBTYPE_RST_SUBSYSTEM(0U)
```

Description

Define PM_SHUTDOWN_SUBTYPE_RST_PS_ONLY

Definition

```
#define PM_SHUTDOWN_SUBTYPE_RST_PS_ONLY(1U)
```

Description

Define PM_SHUTDOWN_SUBTYPE_RST_SYSTEM

Definition

```
#define PM_SHUTDOWN_SUBTYPE_RST_SYSTEM(2U)
```

Description

Define PM_SUSPEND_STATE_CPU_IDLE

Definition

```
#define PM_SUSPEND_STATE_CPU_IDLE 0x0
```

Description

Define PM_SUSPEND_STATE_SUSPEND_TO_RAM

Definition

```
#define PM_SUSPEND_STATE_SUSPEND_TO_RAM 0x1
```

Description

Define XPM_RPU_MODE_LOCKSTEP

Definition

```
#define XPM_RPU_MODE_LOCKSTEP 0x0
```

Description

Define XPM_RPU_MODE_SPLIT

Definition

```
#define XPM_RPU_MODE_SPLIT 0x1
```

Description

Define XPM_RPU_BOOTMEM_LOVEC

Definition

```
#define XPM_RPU_BOOTMEM_LOVEC(OU)
```

Description

Define XPM_RPU_BOOTMEM_HIVEC

Definition

```
#define XPM_RPU_BOOTMEM_HIVEC(1U)
```

Description

Define XPM_RPU_TCM_SPLIT

Definition

```
#define XPM_RPU_TCM_SPLITOU
```

Description

Define XPM_RPU_TCM_COMB

Definition

```
#define XPM_RPU_TCM_COMB1U
```

Description

Define XPM_BOOT_HEALTH_STATUS_MASK

Definition

```
#define XPM_BOOT_HEALTH_STATUS_MASK(0x1U)
```

Description

Define XPM_TAPDELAY_QSPI

Definition

```
#define XPM_TAPDELAY_QSPI(2U)
```

Description

Define XPM_TAPDELAY_BYPASS_DISABLE

Definition

```
#define XPM_TAPDELAY_BYPASS_DISABLE(0U)
```

Description

Define XPM_TAPDELAY_BYPASS_ENABLE

Definition

```
#define XPM_TAPDELAY_BYPASS_ENABLE(1U)
```

Description

Define XPM_OSPI_MUX_SEL_DMA

Definition

```
#define XPM_OSPI_MUX_SEL_DMA(OU)
```

Description

Define XPM_OSPI_MUX_SEL_LINEAR

Definition

```
#define XPM_OSPI_MUX_SEL_LINEAR(1U)
```

Description

Define XPM_OSPI_MUX_GET_MODE

Definition

```
#define XPM_OSPI_MUX_GET_MODE(2U)
```

Description

Define XPM_TAPDELAY_INPUT

Definition

```
#define XPM_TAPDELAY_INPUT(OU)
```

Description

Define XPM_TAPDELAY_OUTPUT

Definition

```
#define XPM_TAPDELAY_OUTPUT(1U)
```

Description

Define XPM_DLL_RESET_ASSERT

Definition

```
#define XPM_DLL_RESET_ASSERT(OU)
```

Description

Define XPM_DLL_RESET_RELEASE

Definition

```
#define XPM_DLL_RESET_RELEASE(1U)
```

Description

Define XPM_DLL_RESET_PULSE

Definition

```
#define XPM_DLL_RESET_PULSE(2U)
```

Description

Define XPM_RESET_REASON_EXT_POR

26uq E66qp9cK

Definition

```
#define XPM_RESET_REASON_EXT_POR(OU)
```

Description

Define XPM_RESET_REASON_SW_POR

Definition

Define XPM_RESET_REASON_DAP_SRST

Definition

```
#define XPM_RESET_REASON_DAP_SRST( 7U)
```

Description

Define XPM_RESET_REASON_ERR_SRST

Definition

```
#define XPM_RESET_REASON_ERR_SRST( 8U)
```

Description

Define XPM_RESET_REASON_SW_SRST

Definition

```
#define XPM_RESET_REASON_SW_SRST( 9U)
```

Description

Define XPM_RESET_REASON_SLR_SRST

Definition

```
#define XPM_RESET_REASON_SLR_SRST( 10U)
```

Description

Define XPM_PROBE_COUNTER_TYPE_STATE_PERIOD

Definition

```
#define XPM_PROBE_COUNTER_TYPE_STATE_PERIOD( 3U)
```

Description

Define XPM_PROBE_COUNTER_TYPE_PORT_SEL

Definition

```
#define XPM_PROBE_COUNTER_TYPE_PORT_SEL( 4U)
```

Description

Define XPM_PROBE_COUNTER_TYPE_SRC

Definition

```
#define XPM_PROBE_COUNTER_TYPE_SRC( 5U)
```

Description

Define XPM_PROBE_COUNTER_TYPE_VAL

Definition

```
#define XPM_PROBE_COUNTER_TYPE_VAL( 6U)
```

Description

Define XST_API_BASE_VERSION

Definition

```
#define XST_API_BASE_VERSION(1U)
```

Description

Define XST_API_QUERY_DATA_VERSION

Definition

```
#define XST_API_QUERY_DATA_VERSION(2U)
```

Description

Define XST_API_REG_NOTIFIER_VERSION

Definition

```
#define XST_API_REG_NOTIFIER_VERSION(2U)
```

Description

Define HOUSECLEAN_DISABLE_DEFAULT_MASK

Definition

```
#define HOUSECLEAN_DISABLE_DEFAULT_MASK(0x0000U)
```

Description

Define HOUSECLEAN_DISABLE_SCAN_CLEAR_MASK

Definition

```
#define HOUSECLEAN_DISABLE_SCAN_CLEAR_MASK(0x0001U)
```

Description

Define HOUSECLEAN_DISABLE_BISR_MASK

Definition

```
#define HOUSECLEAN_DISABLE_BISR_MASK(0x0002U)
```

Description

Define HOUSECLEAN_DISABLE_LBIST_MASK

Definition

```
#define HOUSECLEAN_DISABLE_LBIST_MASK(0x0004U)
```

Description

Define HOUSECLEAN_DISABLE_MBIST_CLEAR_MASK

Definition

```
#define HOUSECLEAN_DISABLE_MBIST_CLEAR_MASK(0x0008U)
```

Description

Define HOUSECLEAN_DISABLE_PL_HC_MASK

Definition

```
#define HOUSECLEAN_DISABLE_PL_HC_MASK( 0x0010U)
```

Description

Power Nodes

Definitions

Define PM_POWER_PMC

Definition

```
#define PM_POWER_PMC( 0x4208001U)
```

Description

Define PM_POWER_LPD

Definition

```
#define PM_POWER_LPD( 0x4210002U)
```

Description

Define PM_POWER_FPD

Definition

```
#define PM_POWER_FPD( 0x420c003U)
```

Description***Define PM_POWER_NOC*****Definition**

```
#define PM_POWER_NOC( 0x4214004U)
```

Description***Define PM_POWER_ME*****Definition**

```
#define PM_POWER_ME( 0x421c005U)
```

Description***Define PM_POWER_PLD*****Definition**

```
#define PM_POWER_PLD( 0x4220006U)
```

Description***Define PM_POWER_CPM*****Definition**

```
#define PM_POWER_CPM( 0x4218007U)
```

Description

Define PM_POWER_PL_SYSMON

Definition

```
#define PM_POWER_PL_SYSMON(0x4208008U)
```

Description

Define PM_POWER_RPU0_0

Definition

```
#define PM_POWER_RPU0_0(0x4104009U)
```

Description

Define PM_POWER_GEM0

Definition

```
#define PM_POWER_GEM0(0x410400aU)
```

Description

Define PM_POWER_GEM1

Definition

```
#define PM_POWER_GEM1(0x410400bU)
```

Description

Define PM_POWER_OCM_0

Definition

```
#define PM_POWER_OCM_0(0x410400cU)
```


Description***Define PM_POWER_OCM_1*****Definition**

```
#define PM_POWER_OCM_1(0x410400dU)
```

Description***Define PM_POWER_OCM_2*****Definition**

```
#define PM_POWER_OCM_2(0x410400eU)
```

Description***Define PM_POWER_OCM_3*****Definition**

```
#define PM_POWER_OCM_3(0x410400fU)
```

Description***Define PM_POWER_TCM_0_A*****Definition**

```
#define PM_POWER_TCM_0_A(0x4104010U)
```

Description

Define PM_POWER_TCM_0_B

Definition

```
#define PM_POWER_TCM_0_B(0x4104011U)
```

Description

Define PM_POWER_TCM_1_A

Definition

```
#define PM_POWER_TCM_1_A(0x4104012U)
```

Description

Define PM_POWER_TCM_1_B

Definition

```
#define PM_POWER_TCM_1_B(0x4104013U)
```

Description

Define PM_POWER_ACPU_0

Definition

```
#define PM_POWER_ACPU_0(0x4104014U)
```

Description

Define PM_POWER_ACPU_1

Definition

```
#define PM_POWER_ACPU_1(0x4104015U)
```

Description

Define PM_POWER_L2_BANK_0

Definition

```
#define PM_POWER_L2_BANK_0(0x4104016U)
```

Description

Reset Nodes

Definitions

Define PM_RST_PMC_POR

Definition

```
#define PM_RST_PMC_POR(0xc30c001U)
```

Description

Define PM_RST_PMC

Definition

```
#define PM_RST_PMC(0xc410002U)
```

Description

Define PM_RST_PS_POR

Definition

```
#define PM_RST_PS_POR(0xc30c003U)
```

Description

Define PM_RST_PL_POR

Definition

```
#define PM_RST_PL_POR(0xc30c004U)
```

Description

Define PM_RST_NOC_POR

Definition

```
#define PM_RST_NOC_POR(0xc30c005U)
```

Description

Define PM_RST_FPD_POR

Definition

```
#define PM_RST_FPD_POR(0xc30c006U)
```

Description

Define PM_RST_ACPU_0_POR

Definition

```
#define PM_RST_ACPU_0_POR(0xc30c007U)
```

Description***Define PM_RST_ACPU_1_POR*****Definition**

```
#define PM_RST_ACPU_1_POR(0xc30c008U)
```

Description***Define PM_RST_OCM2_POR*****Definition**

```
#define PM_RST_OCM2_POR(0xc30c009U)
```

Description***Define PM_RST_PS_S_RST*****Definition**

```
#define PM_RST_PS_S_RST(0xc41000aU)
```

Description***Define PM_RST_PL_S_RST*****Definition**

```
#define PM_RST_PL_S_RST(0xc41000bU)
```

Description

Define PM_RST_NOC

Definition

```
#define PM_RST_NOC(0xc41000cU)
```

Description

Define PM_RST_NPI

Definition

```
#define PM_RST_NPI(0xc41000dU)
```

Description

Define PM_RST_SYS_RST_1

Definition

```
#define PM_RST_SYS_RST_1(0xc41000eU)
```

Description

Define PM_RST_SYS_RST_2

Definition

```
#define PM_RST_SYS_RST_2(0xc41000fU)
```

Description

Define PM_RST_SYS_RST_3

Definition

```
#define PM_RST_SYS_RST_3(0xc410010U)
```

Description***Define PM_RST_FPD*****Definition**

```
#define PM_RST_FPD(0xc410011U)
```

Description***Define PM_RST_PL0*****Definition**

```
#define PM_RST_PL0(0xc410012U)
```

Description***Define PM_RST_PL1*****Definition**

```
#define PM_RST_PL1(0xc410013U)
```

Description***Define PM_RST_PL2*****Definition**

```
#define PM_RST_PL2(0xc410014U)
```

Description

Define PM_RST_PL3

Definition

```
#define PM_RST_PL3(0xc410015U)
```

Description

Define PM_RST_APU

Definition

```
#define PM_RST_APU(0xc410016U)
```

Description

Define PM_RST_ACPU_0

Definition

```
#define PM_RST_ACPU_0(0xc410017U)
```

Description

Define PM_RST_ACPU_1

Definition

```
#define PM_RST_ACPU_1(0xc410018U)
```

Description

Define PM_RST_ACPU_L2

Definition

```
#define PM_RST_ACPU_L2(0xc410019U)
```


Description***Define PM_RST_ACPU_GIC*****Definition**

```
#define PM_RST_ACPU_GIC(0xc41001aU)
```

Description***Define PM_RST_RPU_ISLAND*****Definition**

```
#define PM_RST_RPU_ISLAND(0xc41001bU)
```

Description***Define PM_RST_RPU_AMBA*****Definition**

```
#define PM_RST_RPU_AMBA(0xc41001cU)
```

Description***Define PM_RST_R5_0*****Definition**

```
#define PM_RST_R5_0(0xc41001dU)
```

Description

Define PM_RST_R5_1

Definition

```
#define PM_RST_R5_1(0xc41001eU)
```

Description

Define PM_RST_SYSMON_PMC_SEQ_RST

Definition

```
#define PM_RST_SYSMON_PMC_SEQ_RST(0xc41001fU)
```

Description

Define PM_RST_SYSMON_PMC_CFG_RST

Definition

```
#define PM_RST_SYSMON_PMC_CFG_RST(0xc410020U)
```

Description

Define PM_RST_SYSMON_FPD_CFG_RST

Definition

```
#define PM_RST_SYSMON_FPD_CFG_RST(0xc410021U)
```

Description

Define PM_RST_SYSMON_FPD_SEQ_RST

Definition

```
#define PM_RST_SYSMON_FPD_SEQ_RST(0xc410022U)
```

Description***Define PM_RST_SYSMON_LPD*****Definition**

```
#define PM_RST_SYSMON_LPD(0xc410023U)
```

Description***Define PM_RST_PDMA_RST1*****Definition**

```
#define PM_RST_PDMA_RST1(0xc410024U)
```

Description***Define PM_RST_PDMA_RST0*****Definition**

```
#define PM_RST_PDMA_RST0(0xc410025U)
```

Description***Define PM_RST_ADMA*****Definition**

```
#define PM_RST_ADMA(0xc410026U)
```

Description

Define PM_RST_TIMESTAMP

Definition

```
#define PM_RST_TIMESTAMP(0xc410027U)
```

Description

Define PM_RST_OCM

Definition

```
#define PM_RST_OCM(0xc410028U)
```

Description

Define PM_RST_OCM2_RST

Definition

```
#define PM_RST_OCM2_RST(0xc410029U)
```

Description

Define PM_RST_IPI

Definition

```
#define PM_RST_IPI(0xc41002aU)
```

Description

Define PM_RST_SBI

Definition

```
#define PM_RST_SBI(0xc41002bU)
```

Description***Define PM_RST_LPD*****Definition**

```
#define PM_RST_LPD(0xc41002cU)
```

Description***Define PM_RST_QSPI*****Definition**

```
#define PM_RST_QSPI(0xc10402dU)
```

Description***Define PM_RST_OSPI*****Definition**

```
#define PM_RST_OSPI(0xc10402eU)
```

Description***Define PM_RST_SDIO_0*****Definition**

```
#define PM_RST_SDI_O_0(0xc10402fU)
```

Description

Define PM_RST_SDIO_1

Definition

```
#define PM_RST_SDIO_1(0xc104030U)
```

Description

Define PM_RST_I2C_PMC

Definition

```
#define PM_RST_I2C_PMC(0xc104031U)
```

Description

Define PM_RST_GPIO_PMC

Definition

```
#define PM_RST_GPIO_PMC(0xc104032U)
```

Description

Define PM_RST_GEM_0

Definition

```
#define PM_RST_GEM_0(0xc104033U)
```

Description

Define PM_RST_GEM_1

Definition

```
#define PM_RST_GEM_1(0xc104034U)
```

Description***Define PM_RST_SPARE*****Definition**

```
#define PM_RST_SPARE(0xc104035U)
```

Description***Define PM_RST_USB_0*****Definition**

```
#define PM_RST_USB_0(0xc104036U)
```

Description***Define PM_RST_UART_0*****Definition**

```
#define PM_RST_UART_0(0xc104037U)
```

Description***Define PM_RST_UART_1*****Definition**

```
#define PM_RST_UART_1(0xc104038U)
```

Description

Define PM_RST_SPI_0

Definition

```
#define PM_RST_SPI_0(0xc104039U)
```

Description

Define PM_RST_SPI_1

Definition

```
#define PM_RST_SPI_1(0xc10403aU)
```

Description

Define PM_RST_CAN_FD_0

Definition

```
#define PM_RST_CAN_FD_0(0xc10403bU)
```

Description

Define PM_RST_CAN_FD_1

Definition

```
#define PM_RST_CAN_FD_1(0xc10403cU)
```

Description

Define PM_RST_I2C_0

Definition

```
#define PM_RST_I2C_0(0xc10403dU)
```


Description***Define PM_RST_I2C_1*****Definition**

```
#define PM_RST_I2C_1(0xc10403eU)
```

Description***Define PM_RST_GPIO_LPD*****Definition**

```
#define PM_RST_GPIO_LPD(0xc10403fU)
```

Description***Define PM_RST_TTC_0*****Definition**

```
#define PM_RST_TTC_0(0xc104040U)
```

Description***Define PM_RST_TTC_1*****Definition**

```
#define PM_RST_TTC_1(0xc104041U)
```

Description

Define PM_RST_TTC_2

Definition

```
#define PM_RST_TTC_2(0xc104042U)
```

Description

Define PM_RST_TTC_3

Definition

```
#define PM_RST_TTC_3(0xc104043U)
```

Description

Define PM_RST_SWDT_FPD

Definition

```
#define PM_RST_SWDT_FPD(0xc104044U)
```

Description

Define PM_RST_SWDT_LPD

Definition

```
#define PM_RST_SWDT_LPD(0xc104045U)
```

Description

Define PM_RST_USB

Definition

```
#define PM_RST_USB(0xc104046U)
```

Description***Define PM_RST_DPC*****Definition**

```
#define PM_RST_DPC(0xc208047U)
```

Description***Define PM_RST_PMCDBG*****Definition**

```
#define PM_RST_PMCDBG(0xc208048U)
```

Description***Define PM_RST_DBG_TRACE*****Definition**

```
#define PM_RST_DBG_TRACE(0xc208049U)
```

Description***Define PM_RST_DBG_FPD*****Definition**

```
#define PM_RST_DBG_FPD(0xc20804aU)
```

Description

Define PM_RST_DBG_TSTMP

Definition

```
#define PM_RST_DBG_TSTMP(0xc20804bU)
```

Description

Define PM_RST_RPU0_DBG

Definition

```
#define PM_RST_RPU0_DBG(0xc20804cU)
```

Description

Define PM_RST_RPU1_DBG

Definition

```
#define PM_RST_RPU1_DBG(0xc20804dU)
```

Description

Define PM_RST_HSDP

Definition

```
#define PM_RST_HSDP(0xc20804eU)
```

Description

Define PM_RST_DBG_LPD

Definition

```
#define PM_RST_DBG_LPD(0xc20804fU)
```

Description***Define PM_RST_CPM_POR*****Definition**

```
#define PM_RST_CPM_POR(0xc30c050U)
```

Description***Define PM_RST_CPM*****Definition**

```
#define PM_RST_CPM(0xc410051U)
```

Description***Define PM_RST_CPMDBG*****Definition**

```
#define PM_RST_CPMDBG(0xc208052U)
```

Description***Define PM_RST_PCIE_CFG*****Definition**

```
#define PM_RST_PCIE_CFG(0xc410053U)
```

Description

Define PM_RST_PCIE_CORE0

Definition

```
#define PM_RST_PCIE_CORE0(0xc410054U)
```

Description

Define PM_RST_PCIE_CORE1

Definition

```
#define PM_RST_PCIE_CORE1(0xc410055U)
```

Description

Define PM_RST_PCIE_DMA

Definition

```
#define PM_RST_PCIE_DMA(0xc410056U)
```

Description

Define PM_RST_CMN

Definition

```
#define PM_RST_CMN(0xc410057U)
```

Description

Define PM_RST_L2_0

Definition

```
#define PM_RST_L2_0(0xc410058U)
```

Description***Define PM_RST_L2_1*****Definition**

```
#define PM_RST_L2_1(0xc410059U)
```

Description***Define PM_RST_ADDR_REMAP*****Definition**

```
#define PM_RST_ADDR_REMAP(0xc41005aU)
```

Description***Define PM_RST_CPI0*****Definition**

```
#define PM_RST_CPI0(0xc41005bU)
```

Description***Define PM_RST_CPI1*****Definition**

```
#define PM_RST_CPI1(0xc41005cU)
```

Description

Description***Define PM_CLK_RPU_PLL*****Definition**

```
#define PM_CLK_RPU_PLL(0x8104003U)
```

Description***Define PM_CLK_CPM_PLL*****Definition**

```
#define PM_CLK_CPM_PLL(0x8104004U)
```

Description***Define PM_CLK_NOC_PLL*****Definition**

```
#define PM_CLK_NOC_PLL(0x8104005U)
```

Description***Define PM_CLK_PMC_PRESRC*****Definition**

```
#define PM_CLK_PMC_PRESRC(0x8208007U)
```

Description

Define PM_CLK_PMC_POSTCLK

Definition

```
#define PM_CLK_PMC_POSTCLK(0x8208008U)
```

Description

Define PM_CLK_PMC_PLL_OUT

Definition

```
#define PM_CLK_PMC_PLL_OUT(0x8208009U)
```

Description

Define PM_CLK_PPLL

Definition

```
#define PM_CLK_PPLL(0x820800aU)
```

Description

Define PM_CLK_NOC_PRESRC

Definition

```
#define PM_CLK_NOC_PRESRC(0x820800bU)
```

Description

Define PM_CLK_NOC_POSTCLK

Definition

```
#define PM_CLK_NOC_POSTCLK(0x820800cU)
```

Description***Define PM_CLK_NOC_PLL_OUT*****Definition**

```
#define PM_CLK_NOC_PLL_OUT(0x820800dU)
```

Description***Define PM_CLK_NPLL*****Definition**

```
#define PM_CLK_NPLL(0x820800eU)
```

Description***Define PM_CLK_APU_PRESRC*****Definition**

```
#define PM_CLK_APU_PRESRC(0x820800fU)
```

Description***Define PM_CLK_APU_POSTCLK*****Definition**

```
#define PM_CLK_APU_POSTCLK(0x8208010U)
```

Description

Define PM_CLK_APU_PLL_OUT

Definition

```
#define PM_CLK_APU_PLL_OUT(0x8208011U)
```

Description

Define PM_CLK_APLL

Definition

```
#define PM_CLK_APLL(0x8208012U)
```

Description

Define PM_CLK_RPU_PRESRC

Definition

```
#define PM_CLK_RPU_PRESRC(0x8208013U)
```

Description

Define PM_CLK_RPU_POSTCLK

Definition

```
#define PM_CLK_RPU_POSTCLK(0x8208014U)
```

Description

Define PM_CLK_RPU_PLL_OUT

Definition

```
#define PM_CLK_RPU_PLL_OUT(0x8208015U)
```

Description***Define PM_CLK_RPLL*****Definition**

```
#define PM_CLK_RPLL(0x8208016U)
```

Description***Define PM_CLK_CPM_PRESRC*****Definition**

```
#define PM_CLK_CPM_PRESRC(0x8208017U)
```

Description***Define PM_CLK_CPM_POSTCLK*****Definition**

```
#define PM_CLK_CPM_POSTCLK(0x8208018U)
```

Description***Define PM_CLK_CPM_PLL_OUT*****Definition**

```
#define PM_CLK_CPM_PLL_OUT(0x8208019U)
```

Description

Define PM_CLK_CPLL

Definition

```
#define PM_CLK_CPLL(0x820801aU)
```

Description

Define PM_CLK_PPLL_TO_XPD

Definition

```
#define PM_CLK_PPLL_TO_XPD(0x820801bU)
```

Description

Define PM_CLK_NPLL_TO_XPD

Definition

```
#define PM_CLK_NPLL_TO_XPD(0x820801cU)
```

Description

Define PM_CLK_APLL_TO_XPD

Definition

```
#define PM_CLK_APLL_TO_XPD(0x820801dU)
```

Description

Define PM_CLK_RPLL_TO_XPD

Definition

```
#define PM_CLK_RPLL_TO_XPD(0x820801eU)
```

Description***Define PM_CLK_EFUSE_REF*****Definition**

```
#define PM_CLK_EFUSE_REF(0x820801fU)
```

Description***Define PM_CLK_SYSMON_REF*****Definition**

```
#define PM_CLK_SYSMON_REF(0x8208020U)
```

Description***Define PM_CLK_IRO_SUSPEND_REF*****Definition**

```
#define PM_CLK_IRO_SUSPEND_REF(0x8208021U)
```

Description***Define PM_CLK_USB_SUSPEND*****Definition**

```
#define PM_CLK_USB_SUSPEND(0x8208022U)
```

Description

Define PM_CLK_SWITCH_TIMEOUT

Definition

```
#define PM_CLK_SWITCH_TIMEOUT(0x8208023U)
```

Description

Define PM_CLK_RCLK_PMC

Definition

```
#define PM_CLK_RCLK_PMC(0x8208024U)
```

Description

Define PM_CLK_RCLK_LPD

Definition

```
#define PM_CLK_RCLK_LPD(0x8208025U)
```

Description

Define PM_CLK_WDT

Definition

```
#define PM_CLK_WDT(0x8208026U)
```

Description

Define PM_CLK_TTC0

Definition

```
#define PM_CLK_TTC0(0x8208027U)
```


Description***Define PM_CLK_TTC1*****Definition**

```
#define PM_CLK_TTC1(0x8208028U)
```

Description***Define PM_CLK_TTC2*****Definition**

```
#define PM_CLK_TTC2(0x8208029U)
```

Description***Define PM_CLK_TTC3*****Definition**

```
#define PM_CLK_TTC3(0x820802aU)
```

Description***Define PM_CLK_GEM_TSU*****Definition**

```
#define PM_CLK_GEM_TSU(0x820802bU)
```

Description

Define PM_CLK_GEM_TSU_LB**Definition**

```
#define PM_CLK_GEM_TSU_LB(0x820802cU)
```

Description***Define PM_CLK_MUXED_IRO_DIV2*****Definition**

```
#define PM_CLK_MUXED_IRO_DIV2(0x820802dU)
```

Description***Define PM_CLK_MUXED_IRO_DIV4*****Definition**

```
#define PM_CLK_MUXED_IRO_DIV4(0x820802eU)
```

Description***Define PM_CLK_PSM_REF*****Definition**

```
#define PM_CLK_PSM_REF(0x820802fU)
```

Description***Define PM_CLK_GEM0_RX*****Definition**

```
#define PM_CLK_GEM0_RX(0x8208030U)
```

Description***Define PM_CLK_GEM0_TX*****Definition**

```
#define PM_CLK_GEM0_TX(0x8208031U)
```

Description***Define PM_CLK_GEM1_RX*****Definition**

```
#define PM_CLK_GEM1_RX(0x8208032U)
```

Description***Define PM_CLK_GEM1_TX*****Definition**

```
#define PM_CLK_GEM1_TX(0x8208033U)
```

Description***Define PM_CLK_CPM_CORE_REF*****Definition**

```
#define PM_CLK_CPM_CORE_REF(0x8208034U)
```

Description

Define PM_CLK_CPM_LSBUS_REF**Definition**

```
#define PM_CLK_CPM_LSBUS_REF (0x8208035U)
```

Description***Define PM_CLK_CPM_DBG_REF*****Definition**

```
#define PM_CLK_CPM_DBG_REF (0x8208036U)
```

Description***Define PM_CLK_CPM_AUX0_REF*****Definition**

```
#define PM_CLK_CPM_AUX0_REF (0x8208037U)
```

Description***Define PM_CLK_CPM_AUX1_REF*****Definition**

```
#define PM_CLK_CPM_AUX1_REF (0x8208038U)
```

Description***Define PM_CLK_QSPI_REF*****Definition**

```
#define PM_CLK_QSPI_REF (0x8208039U)
```

Description***Define PM_CLK_OSPI_REF*****Definition**

```
#define PM_CLK_OSPI_REF (0x820803aU)
```

Description***Define PM_CLK_SDIO0_REF*****Definition**

```
#define PM_CLK_SDIO0_REF (0x820803bU)
```

Description***Define PM_CLK_SDIO1_REF*****Definition**

```
#define PM_CLK_SDIO1_REF (0x820803cU)
```

Description***Define PM_CLK_PMC_LSBUS_REF*****Definition**

```
#define PM_CLK_PMC_LSBUS_REF (0x820803dU)
```

Description

Define PM_CLK_I2C_REF**Definition**

```
#define PM_CLK_I2C_REF(0x820803eU)
```

Description***Define PM_CLK_TEST_PATTERN_REF*****Definition**

```
#define PM_CLK_TEST_PATTERN_REF(0x820803fU)
```

Description***Define PM_CLK_DFT_OSC_REF*****Definition**

```
#define PM_CLK_DFT_OSC_REF(0x8208040U)
```

Description***Define PM_CLK_PMC_PLO_REF*****Definition**

```
#define PM_CLK_PMC_PLO_REF(0x8208041U)
```

Description***Define PM_CLK_PMC_PL1_REF*****Definition**

```
#define PM_CLK_PMC_PL1_REF(0x8208042U)
```

Description***Define PM_CLK_PMC_PL2_REF*****Definition**

```
#define PM_CLK_PMC_PL2_REF(0x8208043U)
```

Description***Define PM_CLK_PMC_PL3_REF*****Definition**

```
#define PM_CLK_PMC_PL3_REF(0x8208044U)
```

Description***Define PM_CLK_CFU_REF*****Definition**

```
#define PM_CLK_CFU_REF(0x8208045U)
```

Description***Define PM_CLK_SPARE_REF*****Definition**

```
#define PM_CLK_SPARE_REF(0x8208046U)
```

Description

Define PM_CLK_NPI_REF

Definition

```
#define PM_CLK_NPI_REF(0x8208047U)
```

Description

Define PM_CLK_HSM0_REF

Definition

```
#define PM_CLK_HSM0_REF(0x8208048U)
```

Description

Define PM_CLK_HSM1_REF

Definition

```
#define PM_CLK_HSM1_REF(0x8208049U)
```

Description

Define PM_CLK_SD_DLL_REF

Definition

```
#define PM_CLK_SD_DLL_REF(0x820804aU)
```

Description

Define PM_CLK_FPD_TOP_SWITCH

Definition

```
#define PM_CLK_FPD_TOP_SWITCH(0x820804bU)
```


Description***Define PM_CLK_FPD_LSBUS*****Definition**

```
#define PM_CLK_FPD_LSBUS(0x820804cU)
```

Description***Define PM_CLK_ACPU*****Definition**

```
#define PM_CLK_ACPU(0x820804dU)
```

Description***Define PM_CLK_DBG_TRACE*****Definition**

```
#define PM_CLK_DBG_TRACE(0x820804eU)
```

Description***Define PM_CLK_DBG_FPD*****Definition**

```
#define PM_CLK_DBG_FPD(0x820804fU)
```

Description

Define PM_CLK_LPD_TOP_SWITCH

Definition

```
#define PM_CLK_LPD_TOP_SWITCH(0x8208050U)
```

Description

Define PM_CLK_ADMA

Definition

```
#define PM_CLK_ADMA(0x8208051U)
```

Description

Define PM_CLK_LPD_LSBUS

Definition

```
#define PM_CLK_LPD_LSBUS(0x8208052U)
```

Description

Define PM_CLK_CPU_R5

Definition

```
#define PM_CLK_CPU_R5(0x8208053U)
```

Description

Define PM_CLK_CPU_R5_CORE

Definition

```
#define PM_CLK_CPU_R5_CORE(0x8208054U)
```

Description

Define PM_CLK_CPU_R5_OCM

Definition

```
#define PM_CLK_CPU_R5_OCM(0x8208055U)
```

Description

Define PM_CLK_CPU_R5_OCM2

Definition

```
#define PM_CLK_CPU_R5_OCM2(0x8208056U)
```

Description

Define PM_CLK_I0U_SWITCH

Definition

```
#define PM_CLK_I0U_SWITCH(0x8208057U)
```

Description

Define PM_CLK_GEM0_REF

Definition

```
#define PM_CLK_GEM0_REF(0x8208058U)
```

Description

Define PM_CLK_GEM1_REF

Definition

```
#define PM_CLK_GEM1_REF (0x8208059U)
```

Description

Define PM_CLK_GEM_TSU_REF

Definition

```
#define PM_CLK_GEM_TSU_REF (0x820805aU)
```

Description

Define PM_CLK_USB0_BUS_REF

Definition

```
#define PM_CLK_USB0_BUS_REF (0x820805bU)
```

Description

Define PM_CLK_UART0_REF

Definition

```
#define PM_CLK_UART0_REF (0x820805cU)
```

Description

Define PM_CLK_UART1_REF

Definition

```
#define PM_CLK_UART1_REF (0x820805dU)
```

Description***Define PM_CLK_SPI0_REF*****Definition**

```
#define PM_CLK_SPI0_REF (0x820805eU)
```

Description***Define PM_CLK_SPI1_REF*****Definition**

```
#define PM_CLK_SPI1_REF (0x820805fU)
```

Description***Define PM_CLK_CAN0_REF*****Definition**

```
#define PM_CLK_CAN0_REF (0x8208060U)
```

Description***Define PM_CLK_CAN1_REF*****Definition**

```
#define PM_CLK_CAN1_REF (0x8208061U)
```

Description

Define PM_CLK_I2C0_REF**Definition**

```
#define PM_CLK_I2C0_REF(0x8208062U)
```

Description***Define PM_CLK_I2C1_REF*****Definition**

```
#define PM_CLK_I2C1_REF(0x8208063U)
```

Description***Define PM_CLK_DBG_LPD*****Definition**

```
#define PM_CLK_DBG_LPD(0x8208064U)
```

Description***Define PM_CLK_TIMESTAMP_REF*****Definition**

```
#define PM_CLK_TIMESTAMP_REF(0x8208065U)
```

Description***Define PM_CLK_DBG_TSTMP*****Definition**

```
#define PM_CLK_DBG_TSTMP(0x8208066U)
```

Description***Define PM_CLK_CPM_TOPSW_REF*****Definition**

```
#define PM_CLK_CPM_TOPSW_REF(0x8208067U)
```

Description***Define PM_CLK_USB3_DUAL_REF*****Definition**

```
#define PM_CLK_USB3_DUAL_REF(0x8208068U)
```

Description***Define PM_CLK_REF_CLK*****Definition**

```
#define PM_CLK_REF_CLK(0x830c06aU)
```

Description***Define PM_CLK_PL_ALT_REF_CLK*****Definition**

```
#define PM_CLK_PL_ALT_REF_CLK(0x830c06bU)
```

Description

Define PM_CLK_MUXED_IRO

Definition

```
#define PM_CLK_MUXED_IRO(0x830c06cU)
```

Description

Define PM_CLK_PL_EXT

Definition

```
#define PM_CLK_PL_EXT(0x830c06dU)
```

Description

Define PM_CLK_PL_LB

Definition

```
#define PM_CLK_PL_LB(0x830c06eU)
```

Description

Define PM_CLK_MIO_50_OR_51

Definition

```
#define PM_CLK_MIO_50_OR_51(0x830c06fU)
```

Description

Define PM_CLK_MIO_24_OR_25

Definition

```
#define PM_CLK_MIO_24_OR_25(0x830c070U)
```


Description***Define PM_CLK_EMIO*****Definition**

```
#define PM_CLK_EMIO(0x830c071U)
```

Description***Define PM_CLK_MIO*****Definition**

```
#define PM_CLK_MIO(0x830c072U)
```

Description***Define PM_CLK_PL_PMC_ALT_REF_CLK*****Definition**

```
#define PM_CLK_PL_PMC_ALT_REF_CLK(0x830c076U)
```

Description***Define PM_CLK_PL_LPD_ALT_REF_CLK*****Definition**

```
#define PM_CLK_PL_LPD_ALT_REF_CLK(0x830c077U)
```

Description

Define PM_CLK_PL_FPD_ALT_REF_CLK**Definition**

```
#define PM_CLK_PL_FPD_ALT_REF_CLK(0x830c078U)
```

Description

MIO Nodes

Definitions

Define PM_STMIC_LMIO_0**Definition**

```
#define PM_STMIC_LMIO_0(0x14104001U)
```

Description***Define PM_STMIC_LMIO_1*****Definition**

```
#define PM_STMIC_LMIO_1(0x14104002U)
```

Description***Define PM_STMIC_LMIO_2*****Definition**

```
#define PM_STMIC_LMIO_2(0x14104003U)
```

Description***Define PM_STMIC_LMIO_3*****Definition**

```
#define PM_STMIC_LMIO_3(0x14104004U)
```

Description***Define PM_STMIC_LMIO_4*****Definition**

```
#define PM_STMIC_LMIO_4(0x14104005U)
```

Description***Define PM_STMIC_LMIO_5*****Definition**

```
#define PM_STMIC_LMIO_5(0x14104006U)
```

Description***Define PM_STMIC_LMIO_6*****Definition**

```
#define PM_STMIC_LMIO_6(0x14104007U)
```

Description

Description***Define PM_STMIC_LMIO_12*****Definition**

```
#define PM_STMIC_LMIO_12(0x1410400dU)
```

Description***Define PM_STMIC_LMIO_13*****Definition**

```
#define PM_STMIC_LMIO_13(0x1410400eU)
```

Description***Define PM_STMIC_LMIO_14*****Definition**

```
#define PM_STMIC_LMIO_14(0x1410400fU)
```

Description***Define PM_STMIC_LMIO_15*****Definition**

```
#define PM_STMIC_LMIO_15(0x14104010U)
```

Description

Define PM_STMIC_LMIO_16

Definition

```
#define PM_STMIC_LMIO_16(0x14104011U)
```

Description

Define PM_STMIC_LMIO_17

Definition

```
#define PM_STMIC_LMIO_17(0x14104012U)
```

Description

Define PM_STMIC_LMIO_18

Definition

```
#define PM_STMIC_LMIO_18(0x14104013U)
```

Description

Define PM_STMIC_LMIO_19

Definition

```
#define PM_STMIC_LMIO_19(0x14104014U)
```

Description

Define PM_STMIC_LMIO_20

Definition

```
#define PM_STMIC_LMIO_20(0x14104015U)
```

Description***Define PM_STMIC_LMIO_21*****Definition**

```
#define PM_STMIC_LMIO_21(0x14104016U)
```

Description***Define PM_STMIC_LMIO_22*****Definition**

```
#define PM_STMIC_LMIO_22(0x14104017U)
```

Description***Define PM_STMIC_LMIO_23*****Definition**

```
#define PM_STMIC_LMIO_23(0x14104018U)
```

Description***Define PM_STMIC_LMIO_24*****Definition**

```
#define PM_STMIC_LMIO_24(0x14104019U)
```

Description

Define PM_STMIC_LMIO_25

Definition

```
#define PM_STMIC_LMIO_25(0x1410401aU)
```

Description

Define PM_STMIC_PMIO_0

Definition

```
#define PM_STMIC_PMIO_0(0x1410801bU)
```

Description

Define PM_STMIC_PMIO_1

Definition

```
#define PM_STMIC_PMIO_1(0x1410801cU)
```

Description

Define PM_STMIC_PMIO_2

Definition

```
#define PM_STMIC_PMIO_2(0x1410801dU)
```

Description

Define PM_STMIC_PMIO_3

Definition

```
#define PM_STMIC_PMIO_3(0x1410801eU)
```


Description***Define PM_STMIC_PMIO_4*****Definition**

```
#define PM_STMIC_PMIO_4(0x1410801fU)
```

Description***Define PM_STMIC_PMIO_5*****Definition**

```
#define PM_STMIC_PMIO_5(0x14108020U)
```

Description***Define PM_STMIC_PMIO_6*****Definition**

```
#define PM_STMIC_PMIO_6(0x14108021U)
```

Description***Define PM_STMIC_PMIO_7*****Definition**

```
#define PM_STMIC_PMIO_7(0x14108022U)
```

Description

Define PM_STMIC_PMIO_8

Definition

```
#define PM_STMIC_PMIO_8(0x14108023U)
```

Description

Define PM_STMIC_PMIO_9

Definition

```
#define PM_STMIC_PMIO_9(0x14108024U)
```

Description

Define PM_STMIC_PMIO_10

Definition

```
#define PM_STMIC_PMIO_10(0x14108025U)
```

Description

Define PM_STMIC_PMIO_11

Definition

```
#define PM_STMIC_PMIO_11(0x14108026U)
```

Description

Define PM_STMIC_PMIO_12

Definition

```
#define PM_STMIC_PMIO_12(0x14108027U)
```

Description***Define PM_STMIC_PMIO_13*****Definition**

```
#define PM_STMIC_PMIO_13(0x14108028U)
```

Description***Define PM_STMIC_PMIO_14*****Definition**

```
#define PM_STMIC_PMIO_14(0x14108029U)
```

Description***Define PM_STMIC_PMIO_15*****Definition**

```
#define PM_STMIC_PMIO_15(0x1410802aU)
```

Description***Define PM_STMIC_PMIO_16*****Definition**

```
#define PM_STMIC_PMIO_16(0x1410802bU)
```

Description

Define PM_STMIC_PMIO_17

Definition

```
#define PM_STMIC_PMIO_17(0x1410802cU)
```

Description

Define PM_STMIC_PMIO_18

Definition

```
#define PM_STMIC_PMIO_18(0x1410802dU)
```

Description

Define PM_STMIC_PMIO_19

Definition

```
#define PM_STMIC_PMIO_19(0x1410802eU)
```

Description

Define PM_STMIC_PMIO_20

Definition

```
#define PM_STMIC_PMIO_20(0x1410802fU)
```

Description

Define PM_STMIC_PMIO_21

Definition

```
#define PM_STMIC_PMIO_21(0x14108030U)
```

Description***Define PM_STMIC_PMIO_22*****Definition**

```
#define PM_STMIC_PMIO_22(0x14108031U)
```

Description***Define PM_STMIC_PMIO_23*****Definition**

```
#define PM_STMIC_PMIO_23(0x14108032U)
```

Description***Define PM_STMIC_PMIO_24*****Definition**

```
#define PM_STMIC_PMIO_24(0x14108033U)
```

Description***Define PM_STMIC_PMIO_25*****Definition**

```
#define PM_STMIC_PMIO_25(0x14108034U)
```

Description

Define PM_STMIC_PMIO_26

Definition

```
#define PM_STMIC_PMIO_26(0x14108035U)
```

Description

Define PM_STMIC_PMIO_27

Definition

```
#define PM_STMIC_PMIO_27(0x14108036U)
```

Description

Define PM_STMIC_PMIO_28

Definition

```
#define PM_STMIC_PMIO_28(0x14108037U)
```

Description

Define PM_STMIC_PMIO_29

Definition

```
#define PM_STMIC_PMIO_29(0x14108038U)
```

Description

Define PM_STMIC_PMIO_30

Definition

```
#define PM_STMIC_PMIO_30(0x14108039U)
```

Description***Define PM_STMIC_PMIO_31*****Definition**

```
#define PM_STMIC_PMIO_31(0x1410803aU)
```

Description***Define PM_STMIC_PMIO_32*****Definition**

```
#define PM_STMIC_PMIO_32(0x1410803bU)
```

Description***Define PM_STMIC_PMIO_33*****Definition**

```
#define PM_STMIC_PMIO_33(0x1410803cU)
```

Description***Define PM_STMIC_PMIO_34*****Definition**

```
#define PM_STMIC_PMIO_34(0x1410803dU)
```

Description

Define PM_STMIC_PMIO_35

Definition

```
#define PM_STMIC_PMIO_35(0x1410803eU)
```

Description

Define PM_STMIC_PMIO_36

Definition

```
#define PM_STMIC_PMIO_36(0x1410803fU)
```

Description

Define PM_STMIC_PMIO_37

Definition

```
#define PM_STMIC_PMIO_37(0x14108040U)
```

Description

Define PM_STMIC_PMIO_38

Definition

```
#define PM_STMIC_PMIO_38(0x14108041U)
```

Description

Define PM_STMIC_PMIO_39

Definition

```
#define PM_STMIC_PMIO_39(0x14108042U)
```


Description

Define PM_STMIC_PMIO_40

Definition

```
#define PM_STMIC_PMIO_40(0x14108043U)
```

Description

Define PM_STMIC_PMIO_41

Definition

```
#define PM_STMIC_PMIO_41(0x14108044U)
```

Description

Define PM_STMIC_PMIO_42

Definition

```
#define PM_STMIC_PMIO_42(0x14108045U)
```

Description

Define PM_STMIC_PMIO_43

Definition

```
#define PM_STMIC_PMIO_43(0x14108046U)
```

Description

Define PM_STMIC_PMIO_44

Definition

```
#define PM_STMIC_PMIO_44(0x14108047U)
```

Description

Define PM_STMIC_PMIO_45

Definition

```
#define PM_STMIC_PMIO_45(0x14108048U)
```

Description

Define PM_STMIC_PMIO_46

Definition

```
#define PM_STMIC_PMIO_46(0x14108049U)
```

Description

Define PM_STMIC_PMIO_47

Definition

```
#define PM_STMIC_PMIO_47(0x1410804aU)
```

Description

Define PM_STMIC_PMIO_48

Definition

```
#define PM_STMIC_PMIO_48(0x1410804bU)
```

Description***Define PM_STMIC_PMIO_49*****Definition**

```
#define PM_STMIC_PMIO_49(0x1410804cU)
```

Description***Define PM_STMIC_PMIO_50*****Definition**

```
#define PM_STMIC_PMIO_50(0x1410804dU)
```

Description***Define PM_STMIC_PMIO_51*****Definition**

```
#define PM_STMIC_PMIO_51(0x1410804eU)
```

Description

Device Nodes

Definitions

Define PM_DEV_PLD_0

Definition

```
#define PM_DEV_PLD_0(0x18700000U)
```

Description

Define PM_DEV_PMC_PROC

Definition

```
#define PM_DEV_PMC_PROC(0x18104001U)
```

Description

Define PM_DEV_PSM_PROC

Definition

```
#define PM_DEV_PSM_PROC(0x18108002U)
```

Description

Define PM_DEV_ACPU_0

Definition

```
#define PM_DEV_ACPU_0(0x1810c003U)
```

Description

Define PM_DEV_ACPU_1

Definition

```
#define PM_DEV_ACPU_1(0x1810c004U)
```

Description

Define PM_DEV_RPU0_0

Definition

```
#define PM_DEV_RPU0_0(0x18110005U)
```

Description

Define PM_DEV_RPU0_1

Definition

```
#define PM_DEV_RPU0_1(0x18110006U)
```

Description

Define PM_DEV_OCM_0

Definition

```
#define PM_DEV_OCM_0(0x18314007U)
```

Description

Define PM_DEV_OCM_1

Definition

```
#define PM_DEV_OCM_1(0x18314008U)
```

Description***Define PM_DEV_OCM_2*****Definition**

```
#define PM_DEV_OCM_2(0x18314009U)
```

Description***Define PM_DEV_OCM_3*****Definition**

```
#define PM_DEV_OCM_3(0x1831400aU)
```

Description***Define PM_DEV_TCM_0_A*****Definition**

```
#define PM_DEV_TCM_0_A(0x1831800bU)
```

Description***Define PM_DEV_TCM_0_B*****Definition**

```
#define PM_DEV_TCM_0_B(0x1831800cU)
```

Description

Define PM_DEV_TCM_1_A

Definition

```
#define PM_DEV_TCM_1_A(0x1831800dU)
```

Description

Define PM_DEV_TCM_1_B

Definition

```
#define PM_DEV_TCM_1_B(0x1831800eU)
```

Description

Define PM_DEV_L2_BANK_0

Definition

```
#define PM_DEV_L2_BANK_0(0x1831c00fU)
```

Description

Define PM_DEV_DDR_0

Definition

```
#define PM_DEV_DDR_0(0x18320010U)
```

Description

Define PM_DEV_USB_0

Definition

```
#define PM_DEV_USB_0(0x18224018U)
```

Description***Define PM_DEV_GEM_0*****Definition**

```
#define PM_DEV_GEM_0(0x18224019U)
```

Description***Define PM_DEV_GEM_1*****Definition**

```
#define PM_DEV_GEM_1(0x1822401aU)
```

Description***Define PM_DEV_SPI_0*****Definition**

```
#define PM_DEV_SPI_0(0x1822401bU)
```

Description***Define PM_DEV_SPI_1*****Definition**

```
#define PM_DEV_SPI_1(0x1822401cU)
```

Description

Define PM_DEV_I2C_0

Definition

```
#define PM_DEV_I2C_0(0x1822401dU)
```

Description

Define PM_DEV_I2C_1

Definition

```
#define PM_DEV_I2C_1(0x1822401eU)
```

Description

Define PM_DEV_CAN_FD_0

Definition

```
#define PM_DEV_CAN_FD_0(0x1822401fU)
```

Description

Define PM_DEV_CAN_FD_1

Definition

```
#define PM_DEV_CAN_FD_1(0x18224020U)
```

Description

Define PM_DEV_UART_0

Definition

```
#define PM_DEV_UART_0(0x18224021U)
```

Description***Define PM_DEV_UART_1*****Definition**

```
#define PM_DEV_UART_1(0x18224022U)
```

Description***Define PM_DEV_GPIO*****Definition**

```
#define PM_DEV_GPIO(0x18224023U)
```

Description***Define PM_DEV_TTC_0*****Definition**

```
#define PM_DEV_TTC_0(0x18224024U)
```

Description***Define PM_DEV_TTC_1*****Definition**

```
#define PM_DEV_TTC_1(0x18224025U)
```

Description

Define PM_DEV_TTC_2

Definition

```
#define PM_DEV_TTC_2(0x18224026U)
```

Description

Define PM_DEV_TTC_3

Definition

```
#define PM_DEV_TTC_3(0x18224027U)
```

Description

Define PM_DEV_SWDT_LPD

Definition

```
#define PM_DEV_SWDT_LPD(0x18224028U)
```

Description

Define PM_DEV_SWDT_FPD

Definition

```
#define PM_DEV_SWDT_FPD(0x18224029U)
```

Description

Define PM_DEV_OSPI

Definition

```
#define PM_DEV_OSPI(0x1822402aU)
```

Description***Define PM_DEV_QSPI*****Definition**

```
#define PM_DEV_QSPI (0x1822402bU)
```

Description***Define PM_DEV_GPIO_PMC*****Definition**

```
#define PM_DEV_GPIO_PMC(0x1822402cU)
```

Description***Define PM_DEV_I2C_PMC*****Definition**

```
#define PM_DEV_I2C_PMC(0x1822402dU)
```

Description***Define PM_DEV_SDIO_0*****Definition**

```
#define PM_DEV_SDIO_0(0x1822402eU)
```

Description

Define PM_DEV_SDIO_1

Definition

```
#define PM_DEV_SDIO_1(0x1822402fU)
```

Description

Define PM_DEV_RTC

Definition

```
#define PM_DEV_RTC(0x18224034U)
```

Description

Define PM_DEV_ADMA_0

Definition

```
#define PM_DEV_ADMA_0(0x18224035U)
```

Description

Define PM_DEV_ADMA_1

Definition

```
#define PM_DEV_ADMA_1(0x18224036U)
```

Description

Define PM_DEV_ADMA_2

Definition

```
#define PM_DEV_ADMA_2(0x18224037U)
```

Description***Define PM_DEV_ADMA_3*****Definition**

```
#define PM_DEV_ADMA_3(0x18224038U)
```

Description***Define PM_DEV_ADMA_4*****Definition**

```
#define PM_DEV_ADMA_4(0x18224039U)
```

Description***Define PM_DEV_ADMA_5*****Definition**

```
#define PM_DEV_ADMA_5(0x1822403aU)
```

Description***Define PM_DEV_ADMA_6*****Definition**

```
#define PM_DEV_ADMA_6(0x1822403bU)
```

Description

Define PM_DEV_ADMA_7

Definition

```
#define PM_DEV_ADMA_7(0x1822403cU)
```

Description

Define PM_DEV_IPI_0

Definition

```
#define PM_DEV_IPI_0(0x1822403dU)
```

Description

Define PM_DEV_IPI_1

Definition

```
#define PM_DEV_IPI_1(0x1822403eU)
```

Description

Define PM_DEV_IPI_2

Definition

```
#define PM_DEV_IPI_2(0x1822403fU)
```

Description

Define PM_DEV_IPI_3

Definition

```
#define PM_DEV_IPI_3(0x18224040U)
```

Description***Define PM_DEV_IPI_4*****Definition**

```
#define PM_DEV_IPI_4(0x18224041U)
```

Description***Define PM_DEV_IPI_5*****Definition**

```
#define PM_DEV_IPI_5(0x18224042U)
```

Description***Define PM_DEV_IPI_6*****Definition**

```
#define PM_DEV_IPI_6(0x18224043U)
```

Description***Define PM_DEV_SOC*****Definition**

```
#define PM_DEV_SOC(0x18428044U)
```

Description

Define PM_DEV_DDRMC_0

Definition

```
#define PM_DEV_DDRMC_0(0x18520045U)
```

Description

Define PM_DEV_DDRMC_1

Definition

```
#define PM_DEV_DDRMC_1(0x18520046U)
```

Description

Define PM_DEV_DDRMC_2

Definition

```
#define PM_DEV_DDRMC_2(0x18520047U)
```

Description

Define PM_DEV_DDRMC_3

Definition

```
#define PM_DEV_DDRMC_3(0x18520048U)
```

Description

Define PM_DEV_GT_0

Definition

```
#define PM_DEV_GT_0(0x1862c049U)
```

Description***Define PM_DEV_GT_1*****Definition**

```
#define PM_DEV_GT_1(0x1862c04aU)
```

Description***Define PM_DEV_GT_2*****Definition**

```
#define PM_DEV_GT_2(0x1862c04bU)
```

Description***Define PM_DEV_GT_3*****Definition**

```
#define PM_DEV_GT_3(0x1862c04cU)
```

Description***Define PM_DEV_GT_4*****Definition**

```
#define PM_DEV_GT_4(0x1862c04dU)
```

Description

Define PM_DEV_GT_5

Definition

```
#define PM_DEV_GT_5( 0x1862c04eU)
```

Description

Define PM_DEV_GT_6

Definition

```
#define PM_DEV_GT_6( 0x1862c04fU)
```

Description

Define PM_DEV_GT_7

Definition

```
#define PM_DEV_GT_7( 0x1862c050U)
```

Description

Define PM_DEV_GT_8

Definition

```
#define PM_DEV_GT_8( 0x1862c051U)
```

Description

Define PM_DEV_GT_9

Definition

```
#define PM_DEV_GT_9( 0x1862c052U)
```

Description***Define PM_DEV_GT_10*****Definition**

```
#define PM_DEV_GT_10(0x1862c053U)
```

Description***Define PM_DEV_EFUSE_CACHE*****Definition**

```
#define PM_DEV_EFUSE_CACHE(0x18330054U)
```

Description***Define PM_DEV_AMS_ROOT*****Definition**

```
#define PM_DEV_AMS_ROOT(0x18224055U)
```

Description***Define PM_DEV_AIE*****Definition**

```
#define PM_DEV_AIE(0x18224072U)
```

Description

Define PM_DEV_IPI_PMC

Definition

```
#define PM_DEV_IPI_PMC(0x18224073U)
```

Description

Define PM_DEV_GGS_0

Definition

```
#define PM_DEV_GGS_0(0x18248000U)
```

Description

Define PM_DEV_GGS_1

Definition

```
#define PM_DEV_GGS_1(0x18248001U)
```

Description

Define PM_DEV_GGS_2

Definition

```
#define PM_DEV_GGS_2(0x18248002U)
```

Description

Define PM_DEV_GGS_3

Definition

```
#define PM_DEV_GGS_3(0x18248003U)
```

Description***Define PM_DEV_PGGS_0*****Definition**

```
#define PM_DEV_PGGS_0(0x1824c004U)
```

Description***Define PM_DEV_PGGS_1*****Definition**

```
#define PM_DEV_PGGS_1(0x1824c005U)
```

Description***Define PM_DEV_PGGS_2*****Definition**

```
#define PM_DEV_PGGS_2(0x1824c006U)
```

Description***Define PM_DEV_PGGS_3*****Definition**

```
#define PM_DEV_PGGS_3(0x1824c007U)
```

Description

Define PM_DEV_HB_MON_0

Definition

```
#define PM_DEV_HB_MON_0(0x18250000U)
```

Description

Define PM_DEV_HB_MON_1

Definition

```
#define PM_DEV_HB_MON_1(0x18250001U)
```

Description

Define PM_DEV_HB_MON_2

Definition

```
#define PM_DEV_HB_MON_2(0x18250002U)
```

Description

Define PM_DEV_HB_MON_3

Definition

```
#define PM_DEV_HB_MON_3(0x18250003U)
```

Description

Subsystem Nodes

Definitions

Define PM_SUBSYS_DEFAULT

Definition

```
#define PM_SUBSYS_DEFAULT(0x1c000000U)
```

Description

Define PM_SUBSYS_PMC

Definition

```
#define PM_SUBSYS_PMC(0x1c000001U)
```

Description

Event Node IDs

Definitions

Define XPM_NODETYPE_EVENT_ERROR_PMC_ERR1

Definition

```
#define XPM_NODETYPE_EVENT_ERROR_PMC_ERR1(0x28100000U)
```

Description

Define XPM_NODETYPE_EVENT_ERROR_PMC_ERR2**Definition**

```
#define XPM_NODETYPE_EVENT_ERROR_PMC_ERR2(0x28104000U)
```

Description***Define XPM_NODETYPE_EVENT_ERROR_PSM_ERR1*****Definition**

```
#define XPM_NODETYPE_EVENT_ERROR_PSM_ERR1(0x28108000U)
```

Description***Define XPM_NODETYPE_EVENT_ERROR_PSM_ERR2*****Definition**

```
#define XPM_NODETYPE_EVENT_ERROR_PSM_ERR2(0x2810C000U)
```

Description***Define XPM_NODETYPE_EVENT_ERROR_SW_ERR*****Definition**

```
#define XPM_NODETYPE_EVENT_ERROR_SW_ERR(0x28110000U)
```

Description

Error Event Mask

Error Event Mask for PMC ERR1

Definitions

Define XPM_EVENT_ERROR_MASK_BOOT_CR

Definition

```
#define XPM_EVENT_ERROR_MASK_BOOT_CR(0x00000001U)
```

Description

Define XPM_EVENT_ERROR_MASK_BOOT_NCR

Definition

```
#define XPM_EVENT_ERROR_MASK_BOOT_NCR(0x00000002U)
```

Description

Define XPM_EVENT_ERROR_MASK_FW_CR

Definition

```
#define XPM_EVENT_ERROR_MASK_FW_CR(0x00000004U)
```

Description

Define XPM_EVENT_ERROR_MASK_FW_NCR**Definition**

```
#define XPM_EVENT_ERROR_MASK_FW_NCR( 0x00000008U)
```

Description**Define XPM_EVENT_ERROR_MASK_GSW_CR****Definition**

```
#define XPM_EVENT_ERROR_MASK_GSW_CR( 0x00000010U)
```

Description**Define XPM_EVENT_ERROR_MASK_GSW_NCR****Definition**

```
#define XPM_EVENT_ERROR_MASK_GSW_NCR( 0x00000020U)
```

Description**Define XPM_EVENT_ERROR_MASK_CFU****Definition**

```
#define XPM_EVENT_ERROR_MASK_CFU( 0x00000040U)
```

Description**Define XPM_EVENT_ERROR_MASK_CFRAME****Definition**

```
#define XPM_EVENT_ERROR_MASK_CFRAME( 0x00000080U)
```

Description**Define XPM_EVENT_ERROR_MASK_PMC_PSM_CR****Definition**

```
#define XPM_EVENT_ERROR_MASK_PMC_PSM_CR( 0x00000100U)
```

Description**Define XPM_EVENT_ERROR_MASK_PMC_PSM_NCR****Definition**

```
#define XPM_EVENT_ERROR_MASK_PMC_PSM_NCR( 0x00000200U)
```

Description**Define XPM_EVENT_ERROR_MASK_DDRMB_CR****Definition**

```
#define XPM_EVENT_ERROR_MASK_DDRMB_CR( 0x00000400U)
```

Description**Define XPM_EVENT_ERROR_MASK_DDRMB_NCR****Definition**

```
#define XPM_EVENT_ERROR_MASK_DDRMB_NCR( 0x00000800U)
```

Description

Define XPM_EVENT_ERROR_MASK_NOCTYPE1_CR**Definition**

```
#define XPM_EVENT_ERROR_MASK_NOCTYPE1_CR(0x00001000U)
```

Description**Define XPM_EVENT_ERROR_MASK_NOCTYPE1_NCR****Definition**

```
#define XPM_EVENT_ERROR_MASK_NOCTYPE1_NCR(0x00002000U)
```

Description**Define XPM_EVENT_ERROR_MASK_NOCUSER****Definition**

```
#define XPM_EVENT_ERROR_MASK_NOCUSER(0x00004000U)
```

Description**Define XPM_EVENT_ERROR_MASK_MMCM****Definition**

```
#define XPM_EVENT_ERROR_MASK_MMCM(0x00008000U)
```

Description**Define XPM_EVENT_ERROR_MASK_AIE_CR****Definition**

```
#define XPM_EVENT_ERROR_MASK_AIE_CR(0x00010000U)
```

Description

Define XPM_EVENT_ERROR_MASK_AIE_NCR

Definition

```
#define XPM_EVENT_ERROR_MASK_AIE_NCR( 0x00020000U)
```

Description

Define XPM_EVENT_ERROR_MASK_DDRMC_CR

Definition

```
#define XPM_EVENT_ERROR_MASK_DDRMC_CR( 0x00040000U)
```

Description

Define XPM_EVENT_ERROR_MASK_DDRMC_NCR

Definition

```
#define XPM_EVENT_ERROR_MASK_DDRMC_NCR( 0x00080000U)
```

Description

Define XPM_EVENT_ERROR_MASK_GT_CR

Definition

```
#define XPM_EVENT_ERROR_MASK_GT_CR( 0x00100000U)
```

Description

Define XPM_EVENT_ERROR_MASK_GT_NCR**Definition**

```
#define XPM_EVENT_ERROR_MASK_GT_NCR(0x00200000U)
```

Description**Define XPM_EVENT_ERROR_MASK_PLSMON_CR****Definition**

```
#define XPM_EVENT_ERROR_MASK_PLSMON_CR(0x00400000U)
```

Description**Define XPM_EVENT_ERROR_MASK_PLSMON_NCR****Definition**

```
#define XPM_EVENT_ERROR_MASK_PLSMON_NCR(0x00800000U)
```

Description**Define XPM_EVENT_ERROR_MASK_PLO****Definition**

```
#define XPM_EVENT_ERROR_MASK_PLO(0x01000000U)
```

Description**Define XPM_EVENT_ERROR_MASK_PL1****Definition**

```
#define XPM_EVENT_ERROR_MASK_PL1(0x02000000U)
```

Description**Define XPM_EVENT_ERROR_MASK_PL2****Definition**

```
#define XPM_EVENT_ERROR_MASK_PL2(0x04000000U)
```

Description**Define XPM_EVENT_ERROR_MASK_PL3****Definition**

```
#define XPM_EVENT_ERROR_MASK_PL3(0x08000000U)
```

Description**Define XPM_EVENT_ERROR_MASK_NPIROOT****Definition**

```
#define XPM_EVENT_ERROR_MASK_NPIROOT(0x10000000U)
```

Description**Define XPM_EVENT_ERROR_MASK_SSIT3****Definition**

```
#define XPM_EVENT_ERROR_MASK_SSI T3(0x20000000U)
```

Description

Define XPM_EVENT_ERROR_MASK_SSIT4**Definition**

```
#define XPM_EVENT_ERROR_MASK_SSI T4(0x40000000U)
```

Description**Define XPM_EVENT_ERROR_MASK_SSIT5****Definition**

```
#define XPM_EVENT_ERROR_MASK_SSI T5(0x80000000U)
```

Description**Error Event Mask for PMC ERR2*****Definitions*****Define XPM_EVENT_ERROR_MASK_PMCAPB****Definition**

```
#define XPM_EVENT_ERROR_MASK_PMCAPB(0x00000001U)
```

Description

Define XPM_EVENT_ERROR_MASK_PMCROM**Definition**

```
#define XPM_EVENT_ERROR_MASK_PMCROM(0x00000002U)
```

Description**Define XPM_EVENT_ERROR_MASK_MB_FATAL0****Definition**

```
#define XPM_EVENT_ERROR_MASK_MB_FATAL0(0x00000004U)
```

Description**Define XPM_EVENT_ERROR_MASK_MB_FATAL1****Definition**

```
#define XPM_EVENT_ERROR_MASK_MB_FATAL1(0x00000008U)
```

Description**Define XPM_EVENT_ERROR_MASK_PMCAPAR****Definition**

```
#define XPM_EVENT_ERROR_MASK_PMCAPAR(0x00000010U)
```

Description**Define XPM_EVENT_ERROR_MASK_PMC_CR****Definition**

```
#define XPM_EVENT_ERROR_MASK_PMC_CR(0x00000020U)
```

Description

Define XPM_EVENT_ERROR_MASK_PMC_NCR

Definition

```
#define XPM_EVENT_ERROR_MASK_PMC_NCR( 0x00000040U)
```

Description

Define XPM_EVENT_ERROR_MASK_PMCSMON0

Definition

```
#define XPM_EVENT_ERROR_MASK_PMCSMON0( 0x00000080U)
```

Description

Define XPM_EVENT_ERROR_MASK_PMCSMON1

Definition

```
#define XPM_EVENT_ERROR_MASK_PMCSMON1( 0x00000100U)
```

Description

Define XPM_EVENT_ERROR_MASK_PMCSMON2

Definition

```
#define XPM_EVENT_ERROR_MASK_PMCSMON2( 0x00000200U)
```

Description

Define XPM_EVENT_ERROR_MASK_PMCSMON3

Definition

```
#define XPM_EVENT_ERROR_MASK_PMCSMON3( 0x00000400U)
```

Description

Define XPM_EVENT_ERROR_MASK_PMCSMON4

Definition

```
#define XPM_EVENT_ERROR_MASK_PMCSMON4( 0x00000800U)
```

Description

Define XPM_EVENT_ERROR_MASK_PMCSMON8

Definition

```
#define XPM_EVENT_ERROR_MASK_PMCSMON8( 0x00008000U)
```

Description

Define XPM_EVENT_ERROR_MASK_PMCSMON9

Definition

```
#define XPM_EVENT_ERROR_MASK_PMCSMON9( 0x00010000U)
```

Description**Define XPM_EVENT_ERROR_MASK_CFI****Definition**

```
#define XPM_EVENT_ERROR_MASK_CFI (0x00020000U)
```

Description**Define XPM_EVENT_ERROR_MASK_SEUCRC****Definition**

```
#define XPM_EVENT_ERROR_MASK_SEUCRC(0x00040000U)
```

Description**Define XPM_EVENT_ERROR_MASK_SEUECC****Definition**

```
#define XPM_EVENT_ERROR_MASK_SEUECC(0x00080000U)
```

Description**Define XPM_EVENT_ERROR_MASK_RTCALARM****Definition**

```
#define XPM_EVENT_ERROR_MASK_RTCALARM(0x00400000U)
```

Description

Define XPM_EVENT_ERROR_MASK_NPLL

Definition

```
#define XPM_EVENT_ERROR_MASK_NPLL(0x00800000U)
```

Description

Define XPM_EVENT_ERROR_MASK_PPLL

Definition

```
#define XPM_EVENT_ERROR_MASK_PPLL(0x01000000U)
```

Description

Define XPM_EVENT_ERROR_MASK_CLKMON

Definition

```
#define XPM_EVENT_ERROR_MASK_CLKMON(0x02000000U)
```

Description

Define XPM_EVENT_ERROR_MASK_PMCTO

Definition

```
#define XPM_EVENT_ERROR_MASK_PMCTO(0x04000000U)
```

Description

Define XPM_EVENT_ERROR_MASK_PMCXMPU**Definition**

```
#define XPM_EVENT_ERROR_MASK_PMCXMPU( 0x08000000U)
```

Description**Define XPM_EVENT_ERROR_MASK_PMCXPPU****Definition**

```
#define XPM_EVENT_ERROR_MASK_PMCXPPU( 0x10000000U)
```

Description**Define XPM_EVENT_ERROR_MASK_SSI0****Definition**

```
#define XPM_EVENT_ERROR_MASK_SSI0( 0x20000000U)
```

Description**Define XPM_EVENT_ERROR_MASK_SSI1****Definition**

```
#define XPM_EVENT_ERROR_MASK_SSI1( 0x40000000U)
```

Description

Define XPM_EVENT_ERROR_MASK_SSIT2

Definition

```
#define XPM_EVENT_ERROR_MASK_SSI T2(0x80000000U)
```

Description

Error Event Mask for PSM ERR1

Definitions

Define XPM_EVENT_ERROR_MASK_PS_SW_CR

Definition

```
#define XPM_EVENT_ERROR_MASK_PS_SW_CR(0x00000001U)
```

Description

Define XPM_EVENT_ERROR_MASK_PS_SW_NCR

Definition

```
#define XPM_EVENT_ERROR_MASK_PS_SW_NCR(0x00000002U)
```

Description

Define XPM_EVENT_ERROR_MASK_PSM_B_CR

Definition

```
#define XPM_EVENT_ERROR_MASK_PSM_B_CR(0x00000004U)
```


Description

Define XPM_EVENT_ERROR_MASK_PSM_B_NCR

Definition

```
#define XPM_EVENT_ERROR_MASK_PSM_B_NCR(0x00000008U)
```

Description

Define XPM_EVENT_ERROR_MASK_MB_FATAL

Definition

```
#define XPM_EVENT_ERROR_MASK_MB_FATAL(0x00000010U)
```

Description

Define XPM_EVENT_ERROR_MASK_PSM_CR

Definition

```
#define XPM_EVENT_ERROR_MASK_PSM_CR(0x00000020U)
```

Description

Define XPM_EVENT_ERROR_MASK_PSM_NCR

Definition

```
#define XPM_EVENT_ERROR_MASK_PSM_NCR(0x00000040U)
```

Description

Define XPM_EVENT_ERROR_MASK_OCM_ECC**Definition**

```
#define XPM_EVENT_ERROR_MASK_OCM_ECC( 0x00000080U)
```

Description**Define XPM_EVENT_ERROR_MASK_L2_ECC****Definition**

```
#define XPM_EVENT_ERROR_MASK_L2_ECC( 0x00000100U)
```

Description**Define XPM_EVENT_ERROR_MASK_RPU_ECC****Definition**

```
#define XPM_EVENT_ERROR_MASK_RPU_ECC( 0x00000200U)
```

Description**Define XPM_EVENT_ERROR_MASK_RPU_LS****Definition**

```
#define XPM_EVENT_ERROR_MASK_RPU_LS( 0x00000400U)
```

Description

Define XPM_EVENT_ERROR_MASK_RPU_CCF

Definition

```
#define XPM_EVENT_ERROR_MASK_RPU_CCF (0x00000800U)
```

Description

Define XPM_EVENT_ERROR_MASK_GIC_AXI

Definition

```
#define XPM_EVENT_ERROR_MASK_GIC_AXI (0x00001000U)
```

Description

Define XPM_EVENT_ERROR_MASK_GIC_ECC

Definition

```
#define XPM_EVENT_ERROR_MASK_GIC_ECC (0x00002000U)
```

Description

Define XPM_EVENT_ERROR_MASK_APLL_LOCK

Definition

```
#define XPM_EVENT_ERROR_MASK_APLL_LOCK (0x00004000U)
```

Description

Define XPM_EVENT_ERROR_MASK_RPLL_LOCK**Definition**

```
#define XPM_EVENT_ERROR_MASK_RPLL_LOCK( 0x00008000U)
```

Description**Define XPM_EVENT_ERROR_MASK_CPM_CR****Definition**

```
#define XPM_EVENT_ERROR_MASK_CPM_CR( 0x00010000U)
```

Description**Define XPM_EVENT_ERROR_MASK_CPM_NCR****Definition**

```
#define XPM_EVENT_ERROR_MASK_CPM_NCR( 0x00020000U)
```

Description**Define XPM_EVENT_ERROR_MASK_LPD_APB****Definition**

```
#define XPM_EVENT_ERROR_MASK_LPD_APB( 0x00040000U)
```

Description**Define XPM_EVENT_ERROR_MASK_FPD_APB****Definition**

```
#define XPM_EVENT_ERROR_MASK_FPD_APB( 0x00080000U)
```



Description

Define XPM_EVENT_ERROR_MASK_LPD_PAR

Definition

```
#define XPM_EVENT_ERROR_MASK_LPD_PAR( 0x00100000U)
```

Description

Define XPM_EVENT_ERROR_MASK_FPD_PAR

Definition

```
#define XPM_EVENT_ERROR_MASK_FPD_PAR( 0x00200000U)
```

Description

Define XPM_EVENT_ERROR_MASK_I OU_PAR

Definition

```
#define XPM_EVENT_ERROR_MASK_I OU_PAR( 0x00400000U)
```

Description

Define XPM_EVENT_ERROR_MASK_PSM_PAR

Definition

```
#define XPM_EVENT_ERROR_MASK_PSM_PAR( 0x00800000U)
```

Description

Define XPM_EVENT_ERROR_MASK_LPD_TO

Definition

```
#define XPM_EVENT_ERROR_MASK_LPD_TO( 0x01000000U)
```

Description

Define XPM_EVENT_ERROR_MASK_FPD_TO

Definition

```
#define XPM_EVENT_ERROR_MASK_FPD_TO( 0x02000000U)
```

Description

Define XPM_EVENT_ERROR_MASK_PSM_TO

Definition

```
#define XPM_EVENT_ERROR_MASK_PSM_TO( 0x04000000U)
```

Description

Define XPM_EVENT_ERROR_MASK_XRAM_CR

Definition

```
#define XPM_EVENT_ERROR_MASK_XRAM_CR( 0x08000000U)
```

Description

Define XPM_EVENT_ERROR_MASK_XRAM_NCR**Definition**

```
#define XPM_EVENT_ERROR_MASK_XRAM_NCR( 0x10000000U)
```

Description**Error Event Mask for PSM ERR2*****Definitions*****Define XPM_EVENT_ERROR_MASK_LPD_SWDT****Definition**

```
#define XPM_EVENT_ERROR_MASK_LPD_SWDT( 0x00000001U)
```

Description**Define XPM_EVENT_ERROR_MASK_FPD_SWDT****Definition**

```
#define XPM_EVENT_ERROR_MASK_FPD_SWDT( 0x00000002U)
```

Description**Define XPM_EVENT_ERROR_MASK_LPD_XMPU****Definition**

```
#define XPM_EVENT_ERROR_MASK_LPD_XMPU( 0x00040000U)
```

Description

Define XPM_EVENT_ERROR_MASK_LPD_XPPU

Definition

```
#define XPM_EVENT_ERROR_MASK_LPD_XPPU( 0x00080000U)
```

Description

Define XPM_EVENT_ERROR_MASK_FPD_XMPU

Definition

```
#define XPM_EVENT_ERROR_MASK_FPD_XMPU( 0x00100000U)
```

Description

Error Event Mask for Software error events

Definitions

Define XPM_EVENT_ERROR_MASK_HB_MON_0

Definition

```
#define XPM_EVENT_ERROR_MASK_HB_MON_0( 0x00000001U)
```

Description

Define XPM_EVENT_ERROR_MASK_HB_MON_1

Definition

```
#define XPM_EVENT_ERROR_MASK_HB_MON_1( 0x00000002U)
```


Description**Define XPM_EVENT_ERROR_MASK_HB_MON_2****Definition**

```
#define XPM_EVENT_ERROR_MASK_HB_MON_2( 0x00000004U)
```

Description**Define XPM_EVENT_ERROR_MASK_HB_MON_3****Definition**

```
#define XPM_EVENT_ERROR_MASK_HB_MON_3( 0x00000008U)
```

Description

Library Parameters in MSS File

Note

Data Structure Index

pm_acknowledge

```
typedef struct
{
    volatile u8 received,
    u32 node,
    XStatus status,
    u32 opp,
    volatile bool received,
    enum XPmNodeId node
} pm_acknowledge;
```

Table 139: Structure pm_acknowledge member description

Member	Description
received	Has acknowledge argument been received?
node	Node argument about which the acknowledge is
status	Acknowledged status
opp	Operating point of node in question
received	Has acknowledge argument been received?
node	Node argument about which the acknowledge is

pm_init_suspend

```
typedef struct
{
    volatile u8 received,
    enum XPmSuspendReason reason,
    u32 latency,
    u32 state,
    u32 timeout,
    volatile bool received
} pm_init_suspend;
```

Table 140: Structure pm_init_suspend member description

Member	Description
received	Has init suspend callback been received/handled
reason	Reason of initializing suspend
latency	Maximum allowed latency
state	Targeted sleep/suspend state
timeout	Period of time the client has to response
received	Has init suspend callback been received/handled

XPm_DeviceStatus

```
typedef struct
{
    u32 Status,
    u32 Requirement,
    u32 Usage
} XPm_DeviceStatus;
```

Table 141: Structure XPm_DeviceStatus member description

Member	Description
Status	Device power state
Requirement	Requirements placed on the device by the caller
Usage	Usage info (which subsystem is using the device)

XPm_NodeStatus

[XPm_NodeStatus](#)

```
typedef struct
{
    u32 status,
    u32 requirements,
    u32 usage
} XPm_NodeStatus;
```

Table 142: Structure XPm_NodeStatus member description

Member	Description
status	Node power state
requirements	Current requirements asserted on the node (slaves only)
usage	Usage information (which master is currently using the slave)

XPm_Notifier

[XPm_Notifier](#)

```
typedef struct
{
    void(*const callback)(struct XPm_Notifier *const notifier),
    const u32 node,
    u32 event,
    u32 received_event,
    u32 flags,
    volatile u32 oppoint,
    volatile u32 received,
    struct XPm_Notifier * next,
    enum XPmNodeId node,
    enum XPmNotifyEvent event
} XPm_Notifier;
```

Table 143: Structure XPm_Notifier member description

Member	Description
callback	Custom callback handler to be called when the notification is received. The custom handler would execute from interrupt context, it shall return quickly and must not block! (enables event-driven notifications)
node	Node argument (the node to receive notifications about)

Table 143: Structure XPm_Notifier member description (cont'd)

Member	Description
event	Event argument (the event type to receive notifications about)
received_event	Event received from PLM)
flags	Flags
oppoint	Operating point of node in question. Contains the value updated when the last event notification is received. User shall not modify this value while the notifier is registered.
received	How many times the notification has been received - to be used by application (enables polling). User shall not modify this value while the notifier is registered.
next	Pointer to next notifier in linked list. Must not be modified while the notifier is registered. User shall not ever modify this value.
node	Node argument (the node to receive notifications about)
event	Event argument (the event type to receive notifications about)

Additional Resources and Legal Notices

Xilinx Resources

Documentation Navigator and Design Hubs

docnav

Note

Please Read: Important Legal Notices

AUTOMOTIVE APPLICATIONS DISCLAIMER

Copyright