



Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "<u>Embedded -</u> <u>Microcontrollers</u>"

Details

Product Status	Active
Core Processor	XCore
Core Size	32-Bit 8-Core
Speed	1000MIPS
Connectivity	USB
Peripherals	-
Number of I/O	27
Program Memory Size	-
Program Memory Type	ROMIess
EEPROM Size	-
RAM Size	256K x 8
Voltage - Supply (Vcc/Vdd)	0.95V ~ 3.6V
Data Converters	-
Oscillator Type	External
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	48-UFQFN Exposed Pad
Supplier Device Package	48-UQFN (6x6)
Purchase URL	https://www.e-xfl.com/product-detail/xmos/xu208-256-qf48-i10

Email: info@E-XFL.COM

Address: Room A, 16/F, Full Win Commercial Centre, 573 Nathan Road, Mongkok, Hong Kong

- Ports The I/O pins are connected to the processing cores by Hardware Response ports. The port logic can drive its pins high and low, or it can sample the value on its pins optionally waiting for a particular condition. Section 6.3
- Clock blocks xCORE devices include a set of programmable clock blocks that can be used to govern the rate at which ports execute. Section 6.4
- Memory Each xCORE Tile integrates a bank of SRAM for instructions and data, and a block of one-time programmable (OTP) memory that can be configured for system wide security features. Section 9
- PLL The PLL is used to create a high-speed processor clock given a low speed external oscillator. Section 7
- USB The USB PHY provides High-Speed and Full-Speed, device, host, and on-thego functionality. Data is communicated through ports on the digital node. A library is provided to implement USB device functionality. Section 10
- JTAG The JTAG module can be used for loading programs, boundary scan testing, in-circuit source-level debugging and programming the OTP memory. Section 11

1.1 Software

Devices are programmed using C, C++ or xC (C with multicore extensions). XMOS provides tested and proven software libraries, which allow you to quickly add interface and processor functionality such as USB, Ethernet, PWM, graphics driver, and audio EQ to your applications.

1.2 xTIMEcomposer Studio

The xTIMEcomposer Studio development environment provides all the tools you need to write and debug your programs, profile your application, and write images into flash memory or OTP memory on the device. Because xCORE devices operate deterministically, they can be simulated like hardware within xTIMEcomposer: uniquely in the embedded world, xTIMEcomposer Studio therefore includes a static timing analyzer, cycle-accurate simulator, and high-speed in-circuit instrumentation.

xTIMEcomposer can be driven from either a graphical development environment, or the command line. The tools are supported on Windows, Linux and MacOS X and available at no cost from xmos.com/downloads. Information on using the tools is provided in the xTIMEcomposer User Guide, X3766.

3 Pin Configuration



5 Example Application Diagram



- see Section 10 for details on the USB PHY
- ▶ see Section 12 for details on the power supplies and PCB design

-XMOS



Figure 6: Switch, links and channel ends

and packet switched data can both be supported efficiently. Streams provide the fastest possible data rates between xCORE Tiles (up to 250 MBit/s), but each stream requires a single link to be reserved between switches on two tiles. All packet communications can be multiplexed onto a single link.

Information on the supported routing topologies that can be used to connect multiple devices together can be found in the XS1-U Link Performance and Design Guide, X2999.

7 PLL

The PLL creates a high-speed clock that is used for the switch, tile, and reference clock. The initial PLL multiplication value is shown in Figure 7:

Figure 7: The initial PLL multiplier values

	Oscillator	Tile	PLL Ratio	PLL :	settin	gs
r	Frequency	Frequency		OD	F	R
5	9-25 MHz	144-400 MHz	16	1	63	0

Figure 7 also lists the values of OD, F and R, which are the registers that define the ratio of the tile frequency to the oscillator frequency:

$$F_{core} = F_{osc} \times \frac{F+1}{2} \times \frac{1}{R+1} \times \frac{1}{OD+1}$$

OD, *F* and *R* must be chosen so that $0 \le R \le 63$, $0 \le F \le 4095$, $0 \le OD \le 7$, and $260MHz \le F_{osc} \times \frac{F+1}{2} \times \frac{1}{R+1} \le 1.3GHz$. The *OD*, *F*, and *R* values can be modified by writing to the digital node PLL configuration register.

12

	Feature	Bit	Description
	Disable JTAG	0	The JTAG interface is disabled, making it impossible for the tile state or memory content to be accessed via the JTAG interface.
	Disable Link access	1	Other tiles are forbidden access to the processor state via the system switch. Disabling both JTAG and Link access transforms an xCORE Tile into a "secure island" with other tiles free for non-secure user application code.
	Secure Boot	5	The xCORE Tile is forced to boot from address 0 of the OTP, allowing the xCORE Tile boot ROM to be bypassed (<i>see</i> \S 8).
	Redundant rows	7	Enables redundant rows in OTP.
	Sector Lock 0	8	Disable programming of OTP sector 0.
	Sector Lock 1	9	Disable programming of OTP sector 1.
	Sector Lock 2	10	Disable programming of OTP sector 2.
	Sector Lock 3	11	Disable programming of OTP sector 3.
	OTP Master Lock	12	Disable OTP programming completely: disables updates to all sectors and security register.
	Disable JTAG-OTP	13	Disable all (read & write) access from the JTAG interface to this OTP.
Figure 13: Security		2115	General purpose software accessable security register available to end-users.
register features		3122	General purpose user programmable JTAG UserID code extension.

features

10 USB PHY

X010127,

The USB PHY provides High-Speed and Full-Speed, device, host, and on-the-go functionality. The PHY is configured through a set of peripheral registers (Appendix F), and data is communicated through ports on the digital node. A library, libxud_s.a, is provided to implement USB device functionality.

The USB PHY is connected to the ports on Tile 0 and Tile 1 as shown in Figure 14. When the USB PHY is enabled on Tile 0, the ports shown can on Tile 0 only be used with the USB PHY. When the USB PHY is enabled on Tile 1, then the ports shown can on Tile 1 only be used with the USB PHY. All other IO pins and ports are unaffected. The USB PHY should not be enabled on both tiles.

An external resistor of 43.2 ohm (1% tolerance) should connect USB RTUNE to ground, as close as possible to the device.

Figure 14 shows how two clock blocks can be used to clock the USB ports. One clock block for the TXDATA path, and one clock block for the RXDATA path. Details on how to connect those ports are documented in an application note on USB for xCORE-200.

-XMOS-



Logical Core Requirements 10.1

The XMOS XUD software component runs in a single logical core with endpoint and application cores communicating with it via a combination of channel communication and shared memory variables.

Each IN (host requests data from device) or OUT (data transferred from host to device) endpoint requires one logical core.

JTAG 11

The JTAG module can be used for loading programs, boundary scan testing, incircuit source-level debugging and programming the OTP memory.



Figure 15: ITAG chain

> The JTAG chain structure is illustrated in Figure 15. It comprises a single 1149.1 compliant TAP that can be used for boundary scan of the I/O pins. It has a 4-bit IR and 32-bit DR. It also provides access to a chip TAP that in turn can access the xCORE Tile for loading code and debugging.

13 DC and Switching Characteristics

Symbol	Parameter	MIN	ΤΥΡ	MAX	UNITS	Notes
VDD	Tile DC supply voltage	0.95	1.00	1.05	V	
VDDIO	I/O supply voltage	3.135	3.30	3.465	V	
USB_VDD	USB tile DC supply voltage	0.95	1.00	1.05	V	
VDD33	Peripheral supply	3.135	3.30	3.465	V	
PLL_AVDD	PLL analog supply	0.95	1.00	1.05	V	
Cl	xCORE Tile I/O load capacitance			25	pF	
Та	Ambient operating temperature (Commercial)	0		70	°C	
	Ambient operating temperature (Industrial)	-40		85	°C	
Tj	Junction temperature			125	°C	
Tstg	Storage temperature	-65		150	°C	

13.1 Operating Conditions

Figure 20: Operating conditions

13.2 DC Characteristics, VDDIO=3V3

Symbol	Parameter	MIN	ТҮР	MAX	UNITS	Notes
V(IH)	Input high voltage	2.00		3.60	V	А
V(IL)	Input low voltage	-0.30		0.70	V	А
V(OH)	Output high voltage	2.20			V	B, C
V(OL)	Output low voltage			0.40	V	B, C
I(PU)	Internal pull-up current (Vin=0V)	-100			μA	D
I(PD)	Internal pull-down current (Vin=3.3V)			100	μA	D
I(LC)	Input leakage current	-10		10	μA	

Figure 21: DC characteristics

A All pins except power supply pins.

B All general-purpose I/Os are nominal 4 mA.

C Measured with 4 mA drivers sourcing 4 mA, 8 mA drivers sourcing 8 mA.

D Used to guarantee logic state for an I/O when high impedance. The internal pull-ups/pull-downs should not be used to pull external circuitry. In order to pull the pin to the opposite state, a 4K7 resistor is recommended to overome the internal pull current.



Appendices

A Configuration of the XU208-256-QF48

The device is configured through banks of registers, as shown in Figure 32.



Figure 32: Registers

> The following communication sequences specify how to access those registers. Any messages transmitted contain the most significant 24 bits of the channel-end to which a response is to be sent. This comprises the node-identifier and the channel number within the node. if no response is required on a write operation, supply 24-bits with the last 8-bits set, which suppresses the reply message. Any multi-byte data is sent most significant byte first.

A.1 Accessing a processor status register

The processor status registers are accessed directly from the processor instruction set. The instructions GETPS and SETPS read and write a word. The register number should be translated into a processor-status resource identifier by shifting the register number left 8 places, and ORing it with 0x0B. Alternatively, the functions getps(reg) and setps(reg,value) can be used from XC.

A.2 Accessing an xCORE Tile configuration register

xCORE Tile configuration registers can be accessed through the interconnect using the functions write_tile_config_reg(tileref, ...) and read_tile_config_reg(tile \rightarrow ref, ...), where tileref is the name of the xCORE Tile, e.g. tile[1]. These functions implement the protocols described below.

Instead of using the functions above, a channel-end can be allocated to communicate with the xCORE tile configuration registers. The destination of the channel-end should be set to 0xnnnnC20C where nnnnnn is the tile-identifier.

A write message comprises the following:

29

A write message comprises the following:

control-token	24-bit response	8-bit	8-bit	data	control-token
36	channel-end identifier	register number	size		1

The response to a write message comprises either control tokens 3 and 1 (for success), or control tokens 4 and 1 (for failure).

A read message comprises the following:

control-token	24-bit response	8-bit	8-bit	control-token
37	channel-end identifier	register number	size	1

The response to the read message comprises either control token 3, data, and control-token 1 (for success), or control tokens 4 and 1 (for failure).

B Processor Status Configuration

The processor status control registers can be accessed directly by the processor using processor status reads and writes (use getps(reg) and setps(reg,value) for reads and writes).

Number	Perm	Description
0x00	RW	RAM base address
0x01	RW	Vector base address
0x02	RW	xCORE Tile control
0x03	RO	xCORE Tile boot status
0x05	RW	Security configuration
0x06	RW	Ring Oscillator Control
0x07	RO	Ring Oscillator Value
0x08	RO	Ring Oscillator Value
0x09	RO	Ring Oscillator Value
0x0A	RO	Ring Oscillator Value
0x0C	RO	RAM size
0x10	DRW	Debug SSR
0x11	DRW	Debug SPC
0x12	DRW	Debug SSP
0x13	DRW	DGETREG operand 1
0x14	DRW	DGETREG operand 2
0x15	DRW	Debug interrupt type
0x16	DRW	Debug interrupt data
0x18	DRW	Debug core control
0x20 0x27	DRW	Debug scratch
0x30 0x33	DRW	Instruction breakpoint address
0x40 0x43	DRW	Instruction breakpoint control
0x50 0x53	DRW	Data watchpoint address 1
0x60 0x63	DRW	Data watchpoint address 2
0x70 0x73	DRW	Data breakpoint control register
0x80 0x83	DRW	Resources breakpoint mask
0x90 0x93	DRW	Resources breakpoint value
0x9C 0x9F	DRW	Resources breakpoint control register

-XMOS

Figure 33:

Summary

X010127,

Bits	Perm	Init	Description
31:26	RO	-	Reserved
25:18	RW	0	RGMII TX data delay value (in PLL output cycle increments)
17:9	RW	0	RGMII TX clock divider value. TX clk rises when counter (clocked by PLL output) reaches this value and falls when counter reaches (value»1). Value programmed into this field should be actual divide value required minus 1
8	RW	0	Enable RGMII interface periph ports
7:6	RO	-	Reserved
5	RW	0	Select the dynamic mode (1) for the clock divider when the clock divider is enabled. In dynamic mode the clock divider is only activated when all active threads are paused. In static mode the clock divider is always enabled.
4	RW	0	Enable the clock divider. This divides the output of the PLL to facilitate one of the low power modes.
3	RO	-	Reserved
2	RW		Select between UTMI (1) and ULPI (0) mode.
1	RW		Enable the ULPI Hardware support module
0	RO	-	Reserved

0x02: xCORE Tile control

B.4 xCORE Tile boot status: 0x03

This read-only register describes the boot status of the xCORE tile.

-XMOS"-

	Bits	Perm	Init	Description
	31:24	RO	-	Reserved
	23:16	RO		Processor number.
	15:9	RO	-	Reserved
	8	RO		Overwrite BOOT_MODE.
	7:6	RO	-	Reserved
	5	RO		Indicates if core1 has been powered off
	4	RO		Cause the ROM to not poll the OTP for correct read levels
	3	RO		Boot ROM boots from RAM
). 2	2	RO		Boot ROM boots from JTAG
;	1:0	RO		The boot PLL mode pin value.

0x03 xCORE Tile boot status

0x07:	Bits	Perm	Init	Description
Oscillator	31:16	RO	-	Reserved
Value	15:0	RO	0	Ring oscillator Counter data.

B.8 Ring Oscillator Value: 0x08

This register contains the current count of the xCORE Tile Wire ring oscillator. This value is not reset on a system reset.

0x08 Ring Oscillator Value

)8:	Bits	Perm	Init	Description
or	31:16	RO	-	Reserved
ue	15:0	RO	0	Ring oscillator Counter data.

B.9 Ring Oscillator Value: 0x09

This register contains the current count of the Peripheral Cell ring oscillator. This value is not reset on a system reset.

0x09 Ring Oscillator Value

Bits	Perm	Init	Description
31:16	RO	-	Reserved
15:0	RO	0	Ring oscillator Counter data.

B.10 Ring Oscillator Value: 0x0A

This register contains the current count of the Peripheral Wire ring oscillator. This value is not reset on a system reset.

0x0A: Ring Oscillator Value

:	Bits	Perm	Init	Description
r r	31:16	RO	-	Reserved
2	15:0	RO	0	Ring oscillator Counter data.

-XMOS

B.11 RAM size: 0x0C

The size of the RAM in bytes

B.18 Debug interrupt data: 0x16

On a data watchpoint, this register contains the effective address of the memory operation that triggered the debugger. On a resource watchpoint, it countains the resource identifier.

0x16: Debug interrupt data

16: oug	Bits	Perm	Init	Description
ata	31:0	DRW		Value.

B.19 Debug core control: 0x18

This register enables the debugger to temporarily disable logical cores. When returning from the debug interrupts, the cores set in this register will not execute. This enables single stepping to be implemented.

0x18: Debug core control

Bits	Perm	Init	Description
31:8	RO	-	Reserved
7:0	DRW		1-hot vector defining which threads are stopped when not in debug mode. Every bit which is set prevents the respective thread from running.

B.20 Debug scratch: 0x20 .. 0x27

A set of registers used by the debug ROM to communicate with an external debugger, for example over JTAG. This is the same set of registers as the Debug Scratch registers in the xCORE tile configuration.

0x20 .. 0x27: Debug scratch

ebug	Bits	Perm	Init	Description	
atch	31:0	DRW		Value.	

B.21 Instruction breakpoint address: 0x30 .. 0x33

This register contains the address of the instruction breakpoint. If the PC matches this address, then a debug interrupt will be taken. There are four instruction breakpoints that are controlled individually.

	Bits	Perm	Init	Description
	31:24	CRO		Processor ID of this XCore.
0×00:	23:16	CRO		Number of the node in which this XCore is located.
Device	15:8	CRO		XCore revision.
identification	7:0	CRO		XCore version.

C.2 xCORE Tile description 1: 0x01

This register describes the number of logical cores, synchronisers, locks and channel ends available on this xCORE tile.

Bits	Perm	Init	Description
31:24	CRO		Number of channel ends.
23:16	CRO		Number of the locks.
15:8	CRO		Number of synchronisers.
7:0	RO	-	Reserved

0x01: xCORE Tile description 1

C.3 xCORE Tile description 2: 0x02

This register describes the number of timers and clock blocks available on this xCORE tile.

0x02: xCORE Tile description 2

	Bits	Perm	Init	Description
02.	31:16	RO	-	Reserved
Tile	15:8	CRO		Number of clock blocks.
n 2	7:0	CRO		Number of timers.

C.4 Control PSwitch permissions to debug registers: 0x04

-XMOS-

This register can be used to control whether the debug registers (marked with permission CRW) are accessible through the tile configuration registers. When this bit is set, write -access to those registers is disabled, preventing debugging of the xCORE tile over the interconnect.



	Bits	Perm	Init	Description
	31	CRO		Disables write permission on this register
	30:15	RO	-	Reserved
	14	CRO		Disable access to XCore's global debug
	13	RO	-	Reserved
	12	CRO		lock all OTP sectors
	11:8	CRO		lock bit for each OTP sector
	7	CRO		Enable OTP reduanacy
	6	RO	-	Reserved
	5	CRO		Override boot mode and read boot image from OTP
7.	4	CRO		Disable JTAG access to the PLL/BOOT configuration registers
у. У	3:1	RO	-	Reserved
n	0	CRO		Disable access to XCore's JTAG debug TAP

0x07 Security configuration

C.8 Debug scratch: 0x20 .. 0x27

A set of registers used by the debug ROM to communicate with an external debugger, for example over the switch. This is the same set of registers as the Debug Scratch registers in the processor status.

0x20 .. 0x27: Debug scratch

ebug ratch	Bits	Perm	Init	Description
	31:0	CRW		Value.

C.9 PC of logical core 0: 0x40

Value of the PC of logical core 0.

0x40 PC of logical core 0

ical	Bits	Perm	Init	Description
re 0	31:0	CRO		Value.

-XMOS

C.10 PC of logical core 1: 0x41

Value of the PC of logical core 1.



D Node Configuration

The digital node control registers can be accessed using configuration reads and writes (use write_node_config_reg(device, ...) and read_node_config_reg(device, \rightarrow ...) for reads and writes).

Number	Perm	Description
0x00	RO	Device identification
0x01	RO	System switch description
0x04	RW	Switch configuration
0x05	RW	Switch node identifier
0x06	RW	PLL settings
0x07	RW	System switch clock divider
0x08	RW	Reference clock
0x09	R	System JTAG device ID register
0x0A	R	System USERCODE register
0x0C	RW	Directions 0-7
0x0D	RW	Directions 8-15
0x10	RW	Reserved
0x11	RW	Reserved.
0x1F	RO	Debug source
0x20 0x28	RW	Link status, direction, and network
0x40 0x47	RO	PLink status and network
0x80 0x88	RW	Link configuration and initialization
0xA0 0xA7	RW	Static link configuration

Figure 35: Summary

D.1 Device identification: 0x00

This register contains version and revision identifiers and the mode-pins as sampled at boot-time.

	Bits	Perm	Init	Description
	31:24	RO	-	Reserved
0x00	23:16	RO		Sampled values of BootCtl pins on Power On Reset.
Device	15:8	RO		SSwitch revision.
ification	7:0	RO		SSwitch version.

-XMOS[®]

ident

Bits	Perm	Init	Description
31:5	RO	-	Reserved
4	RW		Reserved.
3:2	RO	-	Reserved
1	RW		If set, XCore1 is the source of last GlobalDebug event.
0	RW		If set, XCore0 is the source of last GlobalDebug event.

0x1F: Debug source

D.15 Link status, direction, and network: 0x20 .. 0x28

These registers contain status information for low level debugging (read-only), the network number that each link belongs to, and the direction that each link is part of. The registers control links 0..7.

Bits	Perm	Init	Description
31:26	RO	-	Reserved
25:24	RO		Identify the SRC_TARGET type 0 - SLink, 1 - PLink, 2 - SSCTL, 3 - Undefine.
23:16	RO		When the link is in use, this is the destination link number to which all packets are sent.
15:12	RO	-	Reserved
11:8	RW	0	The direction that this link operates in.
7:6	RO	-	Reserved
5:4	RW	0	Determines the network to which this link belongs, reset as 0.
3	RO	-	Reserved
2	RO		1 when the current packet is considered junk and will be thrown away.
1	RO		1 when the dest side of the link is in use.
0	RO		1 when the source side of the link is in use.

0x20 .. 0x28: Link status, direction, and network

D.16 PLink status and network: 0x40 .. 0x47

These registers contain status information and the network number that each processor-link belongs to.

E.3 Node identifier: 0x05

0x05: Node identifier

0x51: System clock frequency

Bits	Perm	Init	Description
31:16	RO	-	Reserved
15:0	RW	0	16-bit node identifier. This does not need to be set, and present for compatibility with XS1-switches.

E.4 System clock frequency: 0x51

Bits	Perm	Init	Description
31:7	RO	-	Reserved
6:0	RW	25	Oscillator clock frequency in MHz rounded up to the nearest integer value. Only values between 5 and 100 MHz are valid - writes outside this range are ignored and will be NACKed. This field must be set on start up of the device and any time that the input oscillator clock frequency is changed. It must contain the system clock frequency in MHz rounded up to the nearest integer value.

E.5 Link Control and Status: 0x80

Bits	Perm	Init	Description
31:28	RO	-	Reserved
27	RO		Rx buffer overflow or illegal token encoding received.
26	RO	0	This end of the xlink has issued credit to allow the remote end to transmit
25	RO	0	This end of the xlink has credit to allow it to transmit.
24	WO		Clear this end of the xlink's credit and issue a HELLO token.
23	WO		Reset the receiver. The next symbol that is detected will be the first symbol in a token.
22	RO	-	Reserved
21:11	RW	1	Specify min. number of idle system clocks between two contin- uous symbols witin a transmit token -1.
10:0	RW	1	Specify min. number of idle system clocks between two contin- uous transmit tokens -1.

-XMOS

0x80: Link Control and Status

XS2-U8A-256-QF48

is

0x20: UIFM Sticky flags

Bits	Perm	Init	Description
31:7	RO	-	Reserved
6:0	RW	0	Stickyness for each flag.

F.10 UIFM port masks: 0x24

Set of masks that identify how port 1N, port 1O and port 1P are affected by changes to the flags in FLAGS

Bits	Perm	Init	Description
31:24	RW	0	Bit mask that determines which flags in UIFM_IFM_FLAG[6:0] contribute to port 1?. If any flag listed in this bitmask is high, port 1? will be high.
23:16	RW	0	Bit mask that determines which flags in UIFM_IFM_FLAG[6:0] contribute to port 1P. If any flag listed in this bitmask is high, port 1P will be high.
15:8	RW	0	Bit mask that determines which flags in UIFM_IFM_FLAG[6:0] contribute to port 10. If any flag listed in this bitmask is high, port 10 will be high.
7:0	RW	0	Bit mask that determines which flags in UIFM_IFM_FLAG[6:0] contribute to port 1N. If any flag listed in this bitmask is high, port 1N will be high.

0x24: UIFM port masks

F.11 UIFM SOF value: 0x28

USB Start-Of-Frame counter

0x28: UIFM SOF value

Bits	Perm	Init	Description
31:11	RO	-	Reserved
10:8	RW	0	Most significant 3 bits of SOF counter
7:0	RW	0	Least significant 8 bits of SOF counter

-XMOS

F.12 UIFM PID: 0x2C

The last USB packet identifier received



	Bits	Perm	Init	Description
0x2C:	31:4	RO	-	Reserved
UIFM PID	3:0	RO	0	Value of the last received PID.

F.13 UIFM Endpoint: 0x30

The last endpoint seen

0x30 UIFM Endpoint

	Bits	Perm	Init	Description
-	31:5	RO	-	Reserved
,. 1	4	RO	0	1 if endpoint contains a valid value.
t	3:0	RO	0	A copy of the last received endpoint.

F.14 UIFM Endpoint match: 0x34

This register can be used to mark UIFM endpoints as special.

0x34: UIFM Endpoint match

Bits	Perm	Init	Description
31:16	RO	-	Reserved
15:0	RW	0	This register contains a bit for each endpoint. If its bit is set, the endpoint will be supplied on the RX port when ORed with $0x10$.

F.15 OTG Flags mask: 0x38

0x38: OTG Flags mask

ags ask	Bits	Perm	Init	Description	
	31:0	RW	0	Data	

F.16 UIFM power signalling: 0x3C

	Bits	Perm	Init	Description
0x3C	31:9	RO	-	Reserved
UIFM power	8	RW	0	Valid
signalling	7:0	RW	0	Data