



MKC Michels & Kleberhoff Computer GmbH

Vohwinkeler Str. 58, D-42329 Wuppertal

Tel.: ++49 (0)202 27317 0 Fax: ++49 (0)202 27317 49

Internet: <http://www.mkc-gmbh.de>

Hardware

eWebSrv

Hinweise:

Die Informationen in diesem Handbuch wurden sorgfältig zusammengestellt und überprüft. Dieses Handbuch wird stetig auf dem aktuellen Zustand gehalten. Jedoch wird von **MKC** keine Gewähr für fehlerhafte Informationen übernommen.

MKC behält sich das Recht vor, jederzeit ohne weitere Ankündigung technische Änderungen zur Verbesserung der Zuverlässigkeit, der Funktion oder des Designs der Software und Überarbeitungen des Handbuchs durchzuführen. Änderungen des Handbuchs zwischen 2 Ausgaben werden im Text nicht markiert.

Das Datum einer Ausgabe bezieht sich auf das Handbuch. Dieses muss nicht mit dem Datum der Änderung der Software übereinstimmen. Bei der Versionsgeschichte wird der Grund für die Handbuch Änderungen genannt.

MKC übernimmt keine Haftung für die Anwendung der hier beschriebenen Software. **MKC** übernimmt weiterhin keine Haftung für Schäden oder Folgeschäden, die durch Verwendung dieser Software entstehen. Diese Haftungseinschränkung bezieht sich sowohl auf jeden direkten Abnehmer sowie auf alle seine Kunden und alle Anwender dieser Software.

Mündliche Zusagen über die Anwendbarkeit dieser Software gelten als nicht erfolgt.

Die unten angegebenen Lieferversionen sind zur Zeit verfügbar. Damit ist nicht zugesagt, dass alle diese Versionen weiterhin lieferbar bleiben. **MKC** behält sich das Recht vor, die Produktion dieser Software aus technischen Gründen ohne vorherige Ankündigung einzustellen.

Zahlenangaben:

Hexadezimale Zahlen werden in diesem Handbuch durch ein vorangestelltes Dollarzeichen „\$“ gekennzeichnet. Andere geläufige Schreibweisen für Hexadezimale Zahlen sind z.B. durch den Präfix „0x“ oder den Suffix „h“ in der Literatur angegeben. Sie werden hier nur der Vollständigkeit halber erwähnt.

Um die Lesbarkeit von langen hexadezimalen Zahlen zu verbessern, werden diese von rechts durch einen Punkt in 4er Gruppen unterteilt. Eine mathematische Bedeutung liegt diesem Punkt nicht zugrunde.

Kommentare:

Kommentare oder Korrekturen jedweder Art sind dem Autor jederzeit willkommen. Senden Sie sie bitte an:

MKC Michels & Kleberhoff Computer GmbH
Vohwinkeler Str. 58
42329 Wuppertal

oder

info@mkc-gmbh.de

Handbuch Versionen

Änderungen im Handbuch werden durch eine Erhöhung der Ausgabennummer angezeigt. Handbücher, deren Ausgabe durch einen Buchstaben gekennzeichnet ist, sind vorläufige Handbücher und stimmen möglicherweise noch nicht vollständig mit dem endgültigen Produkt überein. Die erste Ausgabe, die nicht mehr als vorläufig anzusehen ist, beginnt mit der Nummerierung „1“.

Handbuch Versionen			
Ausgabe	Änderungen	Datum	Hardware Revision
A	Übernahme von IONet und eWebSrv Pflichtenheft	14.12.09	
B	FPGA-Schnittstelle SERC3X geändert	09.02.10	
C	Korrekturen, umsortiert	15.03.10	
D	Nur noch ein RUN Bit in der serc3x Schnittstelle	19.03.10	
1	Korrekturen, Übernahme aller Daten	25.03.10	
2	QSPIZBI geändert in QSPIINT	08.04.10	1A/2
3	Korrekturen, Layout aktualisiert	03.08.10	

Lieferversionen ()

Alle Lieferversionen beziehen sich auf den zum Zeitpunkt der Erstellung des Handbuchs gültigen Katalog. Die aktuellen Informationen können Sie unter den obigen Adressen erfahren.

Lieferversionen		
Bestellnummer	Version	Beschreibung
MKC-C-7000		eWebSrv (FPGA: IO48)
MKC-C-7009	eWebTest	Evaluierungssystem für das embedded Modul eWebSrv (FPGA: IO48, EXT: EWEBTEST). Alle erforderlichen Komponenten sind im Lieferumfang enthalten.

Selbstverständlich sind auch Sonderbestückungen, Anpassungen an Ihre Prozessumgebung, usw. nach Absprache möglich. Falls Sie Wünsche, Vorschläge oder kritische Anmerkungen haben, nehmen Sie bitte Kontakt mit uns auf.

Inhaltsverzeichnis

1 EINLEITUNG.....	6
2 TECHNISCHE DATEN.....	7
2.1 Elektrische Eigenschaften.....	8
2.2 Mitgelieferte Hardware und Zubehör.....	9
2.3 Verfügbare Software (www.mkc-gmbh.de)	9
3 HARDWARE SPEZIFIKATION.....	10
3.1 MCF5281.....	10
3.1.1 FLASH.....	11
3.1.2 SRAM.....	11
3.1.3 UART.....	11
3.1.4 Ethernet.....	12
3.1.5 CAN.....	13
3.1.6 GPIO.....	13
3.1.7 QSPI.....	13
3.1.7.1 RTC.....	14
3.1.7.2 Reservierte Pins.....	14
3.1.8 Timer.....	14
3.2 SDRAM.....	14
3.3 FLASH.....	15
3.4 Reset.....	15
3.5 FPGA.....	17
3.5.1 IDREV (fix).....	18
3.5.2 System (fix).....	18
3.5.3 Variable Schnittstelle (OEM-Teil, FPGA).....	19
3.5.3.1 IO48.....	19
3.5.3.2 SERC3X, Serielle Schnittstelle Texas TMS320C3X DSP.....	21
4 STECKERBELEGUNG.....	26
4.1 Pfostenstecker.....	26
4.2 BDM, JTAG.....	27
5 AUSLIEFERZUSTAND.....	28
6 INBETRIEBNAHME.....	28
7 KONFIGURATION, WEB MANAGEMENT INTERFACE.....	28

Liste der Abbildungen

Abbildung 1: Geräteansicht.....	7
Abbildung 2: Blockschaltbild „RESET“.....	16
Abbildung 3: Blockschaltbild FPGA IO48 Pin.....	20
Abbildung 4: Stecker.....	26

Liste der Tabellen

Tabelle 1: Elektrische Eigenschaften.....	8
Tabelle 2: Signale.....	8
Tabelle 3: Interrupt.....	10
Tabelle 4: Speicherbelegung.....	10
Tabelle 5: Speicherbelegung (INTERNES FLASH).....	11
Tabelle 6: UART0 Signale.....	11
Tabelle 7: UART1 Signale.....	11
Tabelle 8: UART Baudraten.....	12
Tabelle 9: ETHERNET Signale.....	12
Tabelle 10: CAN Signale.....	13
Tabelle 11: GPIO Signale.....	13
Tabelle 12: QSPI Signale.....	13
Tabelle 13: RTC Signale.....	14
Tabelle 14: reservierte Signale.....	14
Tabelle 15: TIMER Signale.....	14
Tabelle 16: FLASH SELECT.....	15
Tabelle 17: RESET Signale.....	15
Tabelle 18: FPGA Signale.....	17
Tabelle 19: FPGA JTAG.....	17
Tabelle 20: IDREV REGISTER.....	18
Tabelle 21: SYSTEM REGISTER.....	18
Tabelle 22: Speicherbelegung, IO48.....	19
Tabelle 23: IO48 DATA_IO.....	19
Tabelle 24: IO48 DATA_PIN.....	19
Tabelle 25: IO48 CONFIG_IO.....	20
Tabelle 26: IO48 CONFIG.....	20
Tabelle 27: Speicherbelegung, SERC3X.....	21
Tabelle 28: IRQ REGISTER.....	22
Tabelle 29: Speicherbelegung, SERIAL0.....	22
Tabelle 30: SERIALX, CONTROL.....	23
Tabelle 31: SERIALX, CONTROL.....	24
Tabelle 32: SERIALX, CONTROL.....	25
Tabelle 33: Pin Belegung Pfostenstecker.....	26
Tabelle 34: JTAG/BDM Signale.....	27

1 Einleitung

Unter dem Namen **eWebSrv** stellt **mkc** ein steck- oder lötbare Modul vor, das als Rechneinheit für eine breite Palette von dezentraler, über Netzwerk verfügbarer Prozessperipherie dienen kann.

Im einfachsten Fall ist lediglich eine Trägerkarte nötig, die den **eWebSrv** mit Spannung versorgt und eine RJ45 Buchse enthält.

Zusätzlich kann die Trägerkarte Interfaces für die bis zu 64 GPIOs des **eWebSrv** bereitstellen. Über das Netzwerk können intelligente I/O (z.B. **IONet**) verknüpft werden.

Durch den konsequenten Einsatz dieser dezentralen intelligenten Geräte kann ebenfalls die bisherige kostenintensive Verkabelung von Aktoren und Sensoren mit dem zentralen Steuerrechner entfallen. Die Kommunikation mit den am Prozess installierten Geräten erfolgt über die in jedem Betrieb vorhandene Netzwerk-Infrastruktur (Ethernet, TCP/IP). Unabhängig vom dezentralen Einsatz dieser Geräte kann durch die Speisung über eine zentrale unterbrechungsfreie Stromversorgung (USV) bei einem Stromausfall die gesamte Funktionalität aufrecht erhalten bleiben.

Das prinzipielle Problem von Ethernet bezüglich der Vorhersagbarkeit von Übertragungszeiten (Kollisionen), kann durch den Einsatz von "Switchen" gelöst werden. Diese Verbinden den Absender und den Empfänger einer Verbindung quasi direkt miteinander. Damit steht die volle Übertragungsbandbreite, ohne Unterbrechungen durch andere Teilnehmer, zur Verfügung. Somit können mit dem **eWebSrv** und übergeordneten Systemen (SPS, PC, etc.) vorhersagbare Zykluszeiten im ms Bereich realisiert werden.

Die implementierte Anwenderoberfläche wird mit einem Standard-Browser dargestellt. Dadurch kann der **eWebSrv** von jedem netzwerkfähigen Rechner aus konfiguriert und bedient werden. Alle kritischen Einstellungen werden durch Name und Kennwort abgesichert.

Der Datenaustausch erfolgt über das HTTP-Protokoll (CGI), das TCP/IP-Protokoll oder das UDP/IP-Protokoll. Der TCP- bzw. der UDP-Server senden (einstellbar) periodisch und bei einer Zustandsänderung ein Datenpaket. Die Anbindung an vorhandene Visualisierungen (z.B. LabVIEW) oder eigene Widgets wird somit ohne großen Programmieraufwand möglich.

Entsprechende Beispiele können von unserer Internet-Homepage (www.mkc-gmbh.de) heruntergeladen werden.

Die eindeutige Definition und Offenlegung des Übertragungsverfahrens seitens **mkc** erlaubt es dem Anwender, eigene Übertragungsfunktionen schnell und einfach innerhalb seines Prozesses zu implementieren.

Hinweis: In den folgenden Kapiteln sind Anwahlen in Feldern oder Menüs **fett** und notwendige Eingaben des Benutzers **fett kursiv** angegeben. So ist zum Beispiel die Anwahl des Menüpunktes „Menü1“ und die Eingabe der Zahl 255 im Text folgendermaßen beschrieben: Anwahl **Menü1** und Eingabe **255**.

2 Technische Daten

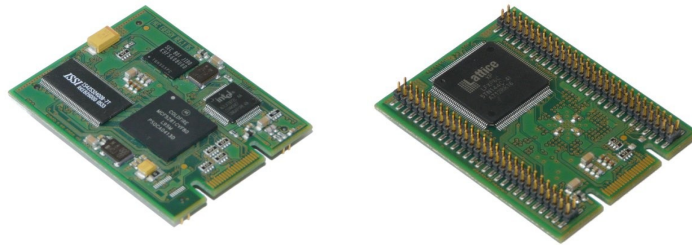


Abbildung 1: Geräteansicht

- Motorola MCF5281 (mit BDM/JTAG Port)
 - 80 MHz
 - 256KB FLASH
 - 64KB SRAM
 - 2 UARTS (RxD/TxD, RTS/CTS)
 - QSPI
 - CAN
 - 16 GPIO
 - 2 TIMER
- 16 MB SDRAM
- 32 MB FLASH (NAND)
- RESET Eingang, RESET Ausgang
- Lattice FPGA LFXP6-5, TQFP144
 - bis zu 48 GPIO
 - weitestgehend frei für Kunden-spezifische Implementierungen
 - flexible Anpassung von 48 Pins der Pfostenleiste
- Abmessungen: T: 47 mm, H: 12 mm, B: 60 mm
- Umgebungstemperatur:
 - Betrieb: 0 ... 50°C
 - Lagerung: -40 ... +85°C
- Kühlung über natürliche Konvektion (kein Lüfter)
- relative Feuchte: 0 ... 90%, nicht kondensierend
- Spannungsversorgung
 - 3,3V (siehe elektrische Eigenschaften)
- Netzwerk
 - 10/100BaseT Ethernet nicht galvanisch getrennt (mit 3 LED Ausgängen)
 - HTTP-Server
 - FTP-Server
 - TELNET-Server
 - Messwertübertragung per TCP/IP, UDP/IP oder HTTP (Intranet / Internet)
- Zentraleinheit für dezentrale Netzwerk Prozessperipherie (z.B. IONet)
- maximal 64 GPIO
 - Messwerterfassung mit 100 Hz Takt
- Nicht flüchtiger Datenspeicher (FLASH) für die Speicherung aller Parameter und der aktuellen Zustände von remanenten Ausgängen
- Betriebssystem uClinux

2.1 Elektrische Eigenschaften

		Min	Typ	Max	Unit
V _{CC}	Supply Voltage	3,15	3,3	3,45	V
I _{CC}	Power Supply Current		600	1000	mA
V _{IH}	High Level Input Voltage	0,7 x V _{CC}		5,25	V
V _{IL}	Low Level Input Voltage			0,35 x V _{CC}	V
V _{OH}	High Level Output Voltage (I _{OH} = -2mA)	V _{CC} - 0,5			V
V _{OL}	Low Level Output Voltage (I _{OL} = 2mA)			0,5	V
V _{IH3}	High Level Input Voltage	2		V _{CC}	V
V _{IL3}	Low Level Input Voltage			0,8	V
V _{OH3}	High Level Output Voltage (I _{OH} = -2mA)	V _{CC} - 0,4			V
V _{OL3}	Low Level Output Voltage (I _{OL} = 2mA)			0,4	V
V _{OHLED}	High Level Output Voltage (I _{OH} = -4mA)	2,4		V _{CC}	V
V _{OLLED}	Low Level Output Voltage (I _{OL} = 4mA)			0,4	V

Tabelle 1: Elektrische Eigenschaften

Eine Beschreibung aller Signale, welche für eine reale Implementierung genutzt werden können, finden Sie in der folgenden Tabelle. Die Signale sind in einzelne Gruppen gegliedert bzw. zusammen gefasst.

Typ		Beschreibung	
I	V _{IH} , V _{IL}	LVTTL 3 Volt Eingang	5V tolerant
IPU	V _{IH} , V _{IL}	LVTTL 3 Volt Eingang	4,7 KΩ PullUp Widerstand 5V tolerant
IPD	V _{IH} , V _{IL}	LVTTL 3 Volt Eingang	4,7 KΩ PullDown Widerstand 5V tolerant
O	V _{OH} , V _{OL}	LVTTL 3 Volt Ausgang	
I3	V _{IH3} , V _{IL3}	LVTTL 3 Volt Eingang	FPGAIO
O3	V _{OH3} , V _{OL3}	LVTTL 3 Volt Ausgang	FPGAIO
OLED	V _{OHLED} , V _{OLLED}	LVTTL 3 Volt Ausgang	PHYLED

Tabelle 2: Signale

2.2 Mitgelieferte Hardware und Zubehör

- Modul eWebSrv

Kundenspezifische Änderungen (OEM) an der Homepage oder am Datenaustausch sind prinzipiell möglich. Das onboard FPGA ist in erster Linie für kundenspezifische Anpassungen vorgesehen.

2.3 Verfügbare Software (www.mkc-gmbh.de)

- Konfigurations-Assistenten (Windows 2000, Windows XP)
- Beispiele zur Messwertübertragung per TCP/IP, UDP/IP oder HTTP (Intranet / Internet)
- Beispiel zur Konfiguration des Netzwerks per UDP-Server

3 Hardware Spezifikation

Wie oben beschrieben handelt es sich beim eWebSrv um ein universelles Modul zum Einsatz auf einer Anwendungsspezifischen Trägerkarte.

Im Folgenden die Beschreibungen der einzelnen Komponenten des Moduls.

3.1 MCF5281

Als Mikroprozessor wird ein MCF528X Coldfire Prozessor der Fa. Freescale eingesetzt. Dieser Baustein hat viele der benötigten Schnittstellen bereits integriert.

Der Prozessor arbeitet mit einer Frequenzen von 80 MHz und ist mit unterschiedlichen internen FLASH Kapazitäten verfügbar.

Hier kommt die Variante MCF5281 zum Einsatz. Sie verfügt über 256KB internes Flash.

Die Interrupts des MCF sind entsprechend der folgenden Tabelle beschaltet.

Interrupt	Quelle
/IRQ1	vorbereitet für QSPI Erweiterungen
/IRQ2	PHY (ETHERNET)
/IRQ3	RTC (QSPI)
/IRQ4	FPGA
/IRQ5	--
/IRQ6	--
/IRQ7	--

Tabelle 3: Interrupt

Der MCF verfügt über einen 32 Bit breit organisierten Adressbereich. Die folgende Tabelle gibt das Anschluss-Schema der einzelnen Baugruppen und deren Speicherbelegung wieder. Für die externen Speicherbereiche werden mit CS Signale generiert, die per Software in den internen Registern des MCF programmiert werden müssen.

CS	Adressen	Baugruppen
--	\$00000000..\$00FFFFFF	16 MB SDRAM (EXTERN)
--	\$20000000..\$2000FFFF	64 KB SRAM (INTERN, RAMBAR)
CS1	\$38000000..\$3BFFFFFF	FPGA INTERFACE (EXTERN, 32 Bit, 4WS)
CS0	\$3C000000..\$3FFFFFFF	NAND FLASH (EXTERN, 8 Bit, 5 WS)
--	\$40000000..\$40FFFFFF	MCF5281 (INTERN, IPSBAR)
--	\$44000000..\$4407FFFF	MCF5281 (INTERN, CFM BACKDOOR)
--	\$F0000000..\$F007FFFF	256 KB FLASH (INTERN, FLASHBAR)

Tabelle 4: Speicherbelegung

3.1.1 FLASH

Das interne FLASH ist z.Zt. in zwei Bereiche aufgeteilt. Der größere Bereich wird vom Linux Loader (mit integriertem Debugger) belegt. Der Zweite beinhaltet Konfiguration, Parameter und remanente Daten des Systems. Eine nähere Beschreibung hierzu ist im Handbuch „System-Programmierer“ zu finden.

Kachel
Coldfire Linux Loader
Konfiguration/Parameter/remanente Daten

Tabelle 5: Speicherbelegung (INTERNES FLASH)

3.1.2 SRAM

Das 64 KB große interne SRAM wird vom Linux Betriebssystem als Pufferspeicher genutzt.

3.1.3 UART

Auf der Baugruppe befinden sich zwei serielle Schnittstellen. Für die Handshake-Leitungen wird das Port TD genutzt. UART0 wird als Konsole des Linux Betriebssystems benutzt.

Serielle Schnittstelle: /Term			
Name	MCF5281	Typ	Beschreibung
/UCTS0	/UCTSx, PTD1, /DIN0	IPU	Status Eingang der seriellen Schnittstelle
/URTS0	/URTSx, PTD3, /DIN1	O	Status Ausgang der seriellen Schnittstelle
UTXD0	PUA0, UTXD0	O	Serieller Datenausgang
URXD0	PUA1, URXD0	IPU	Serieller Dateneingang

Tabelle 6: UART0 Signale

Serielle Schnittstelle: /Aux			
Name	MCF5281	Typ	Beschreibung
/UCTS1	/UCTSx, PTD0, /DTOUT0	IPU	Status Eingang der seriellen Schnittstelle
/URTS1	/URTSx, PTD2, /DTOUT1	O	Status Ausgang der seriellen Schnittstelle
UTXD1	PUA2, UTXD0	O	Serieller Datenausgang
URXD1	PUA3, URXD0	IPU	Serieller Dateneingang

Tabelle 7: UART1 Signale

Jede serielle Schnittstelle kann per Software programmiert werden. Eine nähere Beschreibung hierzu ist im Handbuch „System-Programmierer“ zu finden. Abgeleitet von der oben definierten Frequenz des MCF5281, lassen sich die folgenden Baudraten einstellen.

Baudrate	Teiler	Teiler (Min)			Teiler (Max)		
		Baudrate	Fehler		Baudrate	Fehler	
2400	1041,67				1042	2399,23	-0,03%
4800	520,83				521	4798,46	-0,03%
9600	260,42	260	9615,38	0,16%			
19200	130,21	130	19230,77	0,16%			
38400	65,1	65	38461,54	0,16%			
56000	44,64				45	55555,56	-0,80%
115200	21,7				22	113636,36	-1,38%

Tabelle 8: UART Baudraten

3.1.4 Ethernet

In der CPU ist ein Netzwerk-Controller integriert, welcher noch einen externen PHY benötigt. Hier wird der LTX972ALC PHY von Intel eingesetzt, welcher 100BASE-TX/10BASE-T unterstützt.

Hinweis: Der Takt (25 MHz) für den PHY wird mit einem Oszillator erzeugt.

Der RESET für den PHY wird im FPGA erzeugt und wird per Software gesetzt/rückgesetzt.

7 Signale stehen der Trägerkarte zur Verfügung:

Name	Typ	Beschreibung
RX+ RX-	¹	Twisted Pair Inputs (Positive, Negative)
TX+ TX- TXCP	¹	Twisted Pair Outputs (Positive, Negative, Center)
LED1	OLED	LED Driver, Speed Status
LED2	OLED	LED Driver, Link Status
LED3	OLED	LED Driver, Receive/Transmit Activity

Tabelle 9: ETHERNET Signale

¹ Auf der Trägerkarte müssen nur noch der Übertrager und eine RJ45 Buchse implementiert werden.

3.1.5 CAN

Im Prozessor ist eine CAN Schnittstelle implementiert. Die beiden Signale werden vom Port AS abgegriffen. Die zugehörige Software-Unterstützung ist nicht Implementiert.

Name	MCF5281	Typ	Beschreibung
CANTX	UTXD2, PAS2, CANTX	O	CAN Transmit
CANRX	URXD2, PAS3, CANRX	I	CAN Receive

Tabelle 10: CAN Signale

3.1.6 GPIO

Mehrere Ports des MCF5281 werden als digitale Ein-/Ausgänge benutzt und können per Software programmiert und ausgelesen werden. Insgesamt stehen 16 GPIO Anschlüsse zur freien Verfügung. Eine nähere Beschreibung hierzu ist im Handbuch „System-Programmierer“ zu finden.

Name	MCF5281	Typ	Beschreibung
GPIO[0:3]	PTA[0:3], GPTA[0:3]	I, O	General Purpose I/O
GPIO[4:7]	PTB[0:3], GPTB[0:3]	I, O	General Purpose I/O
GPIO[8:9]	PJ[2:3], CS[2:3]	I, O	General Purpose I/O
GPIO[10:15]	SYNCB, PE0, /TIP SYNCA, PE1, /TS SYNCB, PE2, /SIZ0 SYNCA, PE3, /SIZ1 PE5, /TEA PE7, /OE	I, O	General Purpose I/O

Tabelle 11: GPIO Signale

3.1.7 QSPI

Auf der Baugruppe ist ein serieller SPI Bus (QSPI) definiert, an dem mehrere Baugruppen betrieben werden können.

Name	MCF5281	Typ	Beschreibung
QSPICLK	QSPICLK, PQS0	O	Takt
QSPIDI	QSPIDI, PQS1	I	Daten Eingang
QSPIDO	QSPIDO, PQS2	O	Daten Ausgang
QSPICS[0:3]	QSPICS[0:3], PQS[3:6]	O	^{1,2} 4 Auswahlssignale zur Adressierung

Tabelle 12: QSPI Signale

- ¹ Diese QSPICS Signale (ACTIV-LOW) werden zur direkten Adressierung (CS, Chip Select) von vier Baugruppen am QSPI Bus des MCF5281 benutzt. Die Steuerung erfolgt hierbei über den internen Controller des MCF5281. Um weitere Baugruppen am QSPI Bus betreiben zu können, werden die QSPICS Signale des MCF nicht benutzt. Zur Adressierung (CS, Chip Select) müssen dann zusätzlich Signale benutzt werden. Das Signal QSPICS3 wird für die optionale RTC reserviert. Somit stehen für die direkte Adressierung (MCF5281 Controller) weiterer Baugruppen nur die Signale QSPICS[0:2] zur Verfügung.
- ²

3.1.7.1 RTC

Am QSPI Bus wird eine RTC betrieben (/QSPICS3). Sie ist in der Standard Ausführung des eWebSrv nicht bestückt. Es können Bausteine der Hersteller RICOH (R2045S) und EPSON (RX-4045) bestückt werden.

Der Interrupt der RTC wird mit /IRQ3 des MCF5281 verbunden.

Die RTC wird über den Anschluss EXTBAT mit Spannung versorgt. Auf dem „eWebSrv“ wird die interne Versorgungsspannung (3,3Volt) über eine Schottky-Diode auf EXTBAT gelegt.

Wenn eine externe gepufferte Spannungsversorgung vorgesehen werden soll, ist hier die entsprechende Schaltung für eine Batterie bzw. einen Akku zu realisieren. Weitere Informationen hierzu sind dem Datenblatt der RTC zu entnehmen.

Name	MCF5281	Typ	Beschreibung
EXTBAT	--	--	Externe Spannungsversorgung RTC

Tabelle 13: RTC Signale

3.1.7.2 Reservierte Pins

Im Pfostenverbinder des eWebSrv sind Pins für ein zukünftiges Interface reserviert. Sie sind geplant als zusätzliche Pins für ein QSPI-Device. Es sind vier Signale (an den Ports TC und AS) und ein Interrupt Eingang vorbereitet. Näheres hierzu im Handbuch „System-Programmierer“.

Name	MCF5281	Typ	Beschreibung
QSPIZB0	/UCTSx, PTC1, /DTIN2	I, O	/ATTN
QSPIZB1	/URTSx, PTC3, /DTIN3	I, O	/RST
QSPIZB2	UTXD2, PAS0, SCL	I, O	RXTXEN
QSPIZB3	URXD2, PAS1, SDA	I, O	CE
/QSPIINT	/IRQ1	I	/IRQ

Tabelle 14: reservierte Signale

3.1.8 Timer

Teile des Ports TC der CPU werden als Timerausgänge benutzt und können per Software programmiert werden. Damit stehen 2 Timerausgänge zur Verfügung. Eine nähere Beschreibung hierzu ist im Handbuch „System-Programmierer“ zu finden.

Name	MCF5281	Typ	Beschreibung
TIMER0	/UCTSx, PTC0, /DTOUT2	O	DMA Timer 2
TIMER1	/URTSx, PTC2, /DTOUT3	O	DMA Timer 3

Tabelle 15: TIMER Signale

3.2 SDRAM

Auf dem eWebSrv befindet sich an einer Speicherbank des MCF5281 ein 4x32 MBit SDRAM Chip (z.B. ISSI IS42S32400A-7). Somit stehen 16MB externes RAM zur Verfügung.

3.3 FLASH

Auf dem eWebSrv befindet sich ein 32 MB NAND FLASH im BGA Gehäuse.

Diese Technologie benötigt zusätzliche Signale (kein Adressbus) um Daten schreiben bzw. lesen zu können. Um diese Signale zu setzen oder zu lesen, werden die folgenden MCF5281 Signale verwendet. Eine nähere Beschreibung hierzu ist im Handbuch „System-Programmierer“ zu finden.

Name	MCF5281	Typ	Beschreibung
FL_RB	MA0, PQA0, AN52	I	¹ RB, READY/BUSY OUTPUT
FL_CLE	MA1, PQA1, AN53	O	¹ CLE, COMMAND LATCH ENABLE
FL_ALE	ETRIG1, PQA3, AN55	O	¹ ALE, ADDRESS LATCH ENABLE
/FL_CE	ETRIG2, PQA4, AN46	O	¹ /CE, CHIP ENABLE

Tabelle 16: FLASH SELECT

¹ Diese Signale sind mit den entsprechenden Anschlüssen des NAND FLASH verbunden.

Hinweis: Der Zugriff auf die Daten erfolgt über „CS0“ des MCF. Die zusätzlich vom NAND FLASH benötigten Signale (/RE, /WE) werden im FPGA erzeugt.

Im externen FLASH befindet sich das Linux Betriebssystem. Es wird vom einem Loader im internen FLASH geladen und gestartet. Danach arbeitet auf dem eWebSrv ein 'normales' Linux (uClinux) mit einem Filesystem wie auf einer Festplatte. Genauereres hierzu ist im Handbuch „System-Programmierer“ zu finden.

3.4 Reset

Name	MCF5281	Typ	Beschreibung
/RSTIN	/RSTI	IPU	Mit diesem Eingang kann ein RESET der Baugruppe ausgelöst werden.
/RSTOUT	--	O	/RSTOUT wird aktiv (LOW), sobald eine der unten beschriebenen RESET Quellen aktiv ist. Das Signal kann, nachdem die Baugruppe neu hochgefahren ist per Software deaktiviert werden (HIGH).

Tabelle 17: RESET Signale

Hinweis: Das Signal /RSTOUT wird im FPGA erzeugt.

Auf der Baugruppe existieren mehrere Quellen, welche einen RESET des Systems veranlassen können:

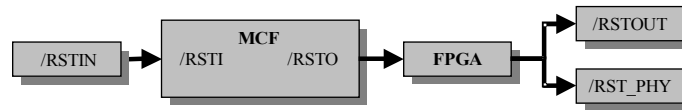


Abbildung 2: Blockschaltbild „RESET“

- Der MCF5281 verfügt über mehrere interne Quellen, welche einen RESET auslösen können. Nähere Informationen stehen im Handbuch (RESET CONTROLLER MODULE).
- Es kann ein RESET ausgelöst werden, indem das Signal /RSTIN für mindestens 400 ns auf „LOW“ gelegt wird.

Nachdem alle RESET Quellen (/RSTIN = HIGH) inaktiv sind, wird das System neu hochgefahren. Wenn die CPU initialisiert ist, wird das Signal /RSTO der CPU deaktiviert (HIGH, RESET CONTROLLER MODULE). Wenn das System ohne Fehler gestartet ist, kann /RSTOUT per Software deaktiviert (HIGH) werden.

Eine nähere Beschreibung wie /RSTOUT zu setzen ist, ist im Handbuch „System-Programmierer“ zu finden. Nach einem RESET wird der Mikroprozessor mit den folgenden Grundeinstellungen initialisiert:

- MASTER MODE, INTERN BOOT, FULL DRIVE STRENGTH, A[23:21]

Diese Einstellungen werden von der Hardware (FPGA) vorgenommen. Nähere Informationen hierzu sind dem Datenblatt des MCF5281 zu entnehmen.

3.5 FPGA

Diese Schnittstelle wird durch ein Lattice LFXP6-5 FPGA in einem TQFP144 realisiert. In diesem Gehäuse stehen 100 I/O Anschlüsse zur Verfügung.

Das FPGA wird für zwei Teilbereiche in der Hardware benutzt:

1. Einen festen Teil, welcher in allen Implementierungen realisiert werden muss. Dieser erzeugt den /RSTPHY, das Signal /RSTOUT (Reset-Erweiterung) und kodiert die Zugriffe auf das externe FLASH (CS0, /FL_RE, /FL_WE).
2. Der zweite Teil ist frei für beliebige Erweiterungen. Dieser Teil stellt 48 frei programmierbare Signale zur Verfügung, die nach Kundenwunsch programmiert werden können. In der Standard Ausführung sind die Pins mit 48 GPIO belegt.

Die I/O Anschlüsse des FPGA sind 3,3 Volt kompatibel.

FPGA	MCF5281	Beschreibung
MCF_D	D[31:0]	32 Bit Datenbus
MCF_A	A[14:9]	6 Adressen
MCF_CLKOUT	CLKOUT	80 MHz Systemtakt
MCF_CS	CS[1:0]	CS Speicherbereiche MCF5281
MCF_IRQ	IRQ4	Interrupt
MCF_RW	R/W	Schreib-/Lesezugriff
MCF_BS	BS[3:0]	4 Signale um 8, 16 oder 32 Bit Zugriffe zu ermöglichen.
/RSTPHY	--	Reset PHY
/RSTOUT	--	Reset Erweiterung
MCF_RSTO	/RSTO	Reset Ausgang MCF
FPGAIO[47:0]		48 I/O Signale, welche je nach programmiertem Inhalt des FPGA genutzt werden können. Der Signaltyp dieser Signale ist I3 bzw. O3

Tabelle 18: FPGA Signale

Das FPGA enthält ein internes FLASH, welches die gültige Implementierung der Hardware enthält. Dieses wird bei der Produktion (JTAG, JTAG/BDM-Schnittstelle) programmiert. Falls erforderlich, kann die Programmierung per Software (JTAG) überschrieben werden. Hierfür werden die folgenden MCF5281 Signale verwendet. Eine nähere Beschreibung hierzu ist im Handbuch „System-Programmierer“ zu finden.

FPGA	MCF5281	Typ	Beschreibung
TDO	ANW, PQB0, AN0	I	JTAG Datenausgang
TDI	ANX, PQB1, AN1	O	JTAG Dateneingang
TCK	ANY, PQB2, AN2	O	JTAG Takteingang
TMS	ANZ, PQB3, AN3	O	JTAG „Mode Select“ Eingang

Tabelle 19: FPGA JTAG

3.5.1 IDREV (fix)

Dieses Register enthält die ID und die Revisionsnummer dieser Schnittstelle. Die ID ist eine eindeutige Nummer, welche den Aufbau dieser Schnittstelle definiert.

Dieses Register gehört zum „festen“ Teil des FPGA und muss in jeder Implementierung definiert sein.

ADDR	R/W	Bit	Name	Reset	
\$00xx	R	[15:0]	ID	ID	
	R	[31:16]	REVISION	REVISION	

Tabelle 20: IDREV REGISTER

3.5.2 System (fix)

Dieses Register steuert Systemfunktionen und gehört zum „festen“ Teil des FPGA und muss in jeder Implementierung definiert sein.

ADDR	R/W	Bit	Name	Reset	ERWEITERUNG
\$02xx	R/W	[0]	RSTOUT	0	Signal /RSTOUT
	R/W	[1]	RSTPHY	0	Signal /RSTPHY
	R	[31:2]	---	0	

Tabelle 21: SYSTEM REGISTER

3.5.3 Variable Schnittstelle (OEM-Teil, FPGA)

Wie bereits erwähnt, ist dieser Teil des FPGA frei für kundenspezifische Implementierungen. Zur Zeit gibt es zwei Varianten. Die Standard-Variante (IO48) bildet jeden der 48 freien Pins des FPGA auf einen digitalen Ein-/Ausgang ab. Die Variante SERC3X stellt 8 serielle Ports an den 48 freien Pins des FPGA zur Verfügung.

3.5.3.1 IO48

Diese Schnittstelle realisiert aus den FPGA Signalen 48 konfigurierbare digitale Ein-/Ausgänge. Die folgende Tabelle gibt das Anschluss-Schemata der einzelnen Baugruppen und deren Speicherbelegung wieder.

CS	Adressen	Baugruppen
CS1	\$38000000..\$3BFFFFFF	FPGA INTERFACE
	\$00xx	IDREV fix
	\$02xx	SYSTEM fix
	\$04xx, \$06xx	DATA_IO
	\$08xx, \$0axx	DATA_PIN
	\$0cxx, \$0dxx	CONFIG_IO
	\$10xx, \$12xx	CONFIG_PIN

Tabelle 22: Speicherbelegung, IO48

DATA_IO wird aus zwei 32 Bit Registern gebildet.

Bei einem Schreibzugriff, wird das interne Register beschrieben. Ist das zugehörige I/O Signal ein Ausgang wird der Ausgang entsprechend gesetzt. Ein Lesezugriff liest den Wert des internen Registers

ADDR	R/W	Bit	Name	Reset	ERWEITERUNG
\$04xx	R/W	[31:0]	IO[31:0]	0	FPGAIO[31:0]
\$06xx	R/W	[15:0]	IO[47:32]	0	FPGAIO[47:32]
	R/W	[31:16]	--	--	--

Tabelle 23: IO48 DATA_IO

DATA_PIN wird aus zwei 32 Bit Registern gebildet.

Bei einem Lesezugriff wird der Zustand am PIN des FPGA (nicht das interne Register) gelesen. Ein Schreibzugriff hat keine Auswirkung.

ADDR	R/W	Bit	Name	Reset	ERWEITERUNG
\$08xx	R	[31:0]	IO[31:0]	--	FPGAIO[31:0]
\$0axx	R	[15:0]	IO[47:32]	--	FPGAIO[47:32]
	R	[31:16]	--	--	--

Tabelle 24: IO48 DATA_PIN

CONFIG_IO wird aus zwei 32 Bit Registern gebildet.

Bei einem Schreibzugriff, wird das interne Register beschrieben und der zugehörige I/O Anschluss konfiguriert (0:Eingang, 1: Ausgang). Ein Lesezugriff liest den Wert des internen Registers

ADDR	R/W	Bit	Name	Reset	ERWEITERUNG
\$0c00	R/W	[31:0]	IO[31:0]	0	FPGAIO[31:0]
\$0d00	R/W	[15:0]	IO[47:32]	0	FPGAIO[47:32]
	R/	[31:16]	--	--	--

Tabelle 25: IO48 CONFIG_IO

CONFIG_PIN wird aus zwei 32 Bit Registern gebildet.

Bei einem Schreibzugriff, wird das interne Register beschrieben und der zugehörige I/O Anschluß konfiguriert (0:Ausgang HIGH-LOW, 1: Open-Kollektor-Ausgang LOW-TRISTATE), wenn er im CONFIG_IO als Ausgang definiert worden ist. Ein Lesezugriff liest den Wert des internen Registers

ADDR	R/W	Bit	Name	Reset	ERWEITERUNG
\$10xx	R/W	[31:0]	IO[31:0]	0	FPGAIO[31:0]
\$12xx	R/W	[15:0]	IO[47:32]	0	FPGAIO[47:32]
	R/	[31:16]	--	--	--

Tabelle 26: IO48 CONFIG

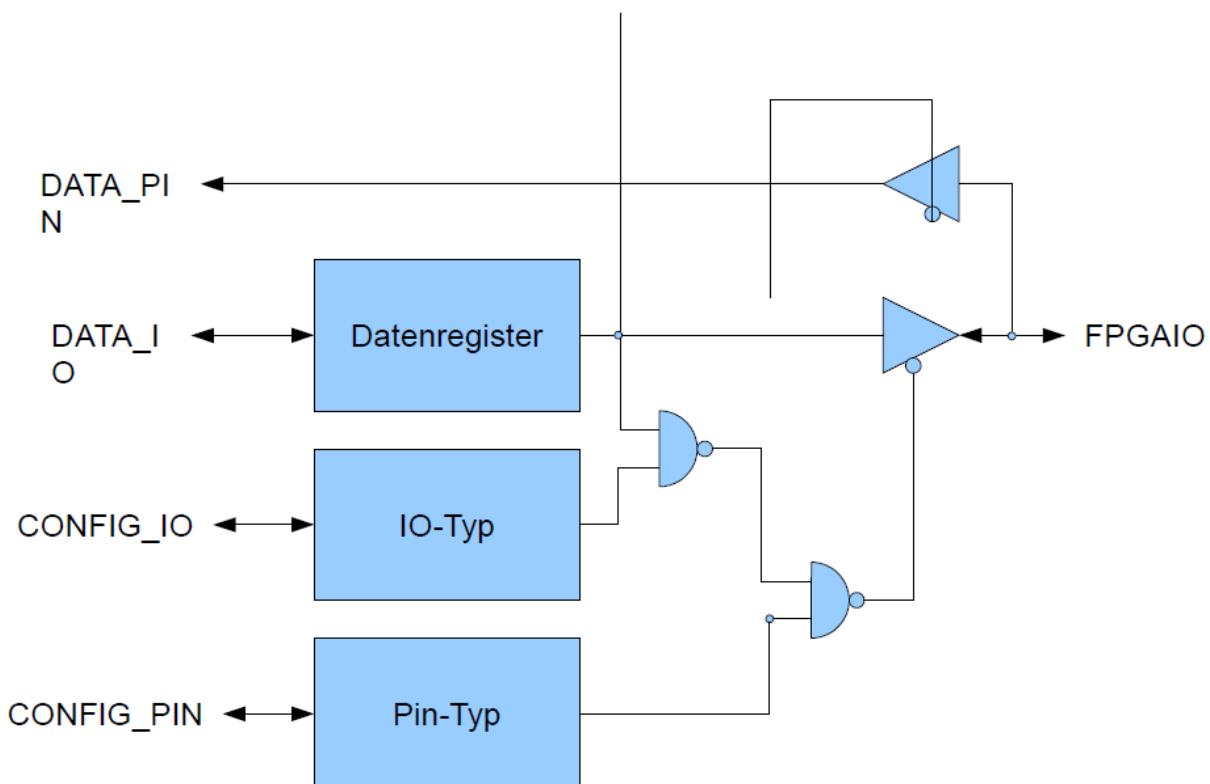


Abbildung 3: Blockschaltbild FPGA IO48 Pin

3.5.3.2 SERC3X, Serielle Schnittstelle Texas TMS320C3X DSP

Hier wird die Logik des FPGA beschrieben. Die benötigte Software ist nicht Bestandteil dieser Definition.

Diese Schnittstelle realisiert aus den FPGA Signalen 8 konfigurierbare serielle Schnittstellen für Texas Instruments TMS320C3X DSPs (TMS320C32, TMS320VC33). Hinweise hierzu sind dem „User's Guide TMS320C3x“ zu entnehmen.

Die Schnittstelle wird im „standard mode“ betrieben. Dies entspricht im „User's Guide TMS320C3x“ der seriellen Schnittstelle „XFSM=0“ des „Serial Global Control Register“.

Alle Register sind 32 Bit breit.

Die folgende Tabelle gibt das Anschluss-Schemata der einzelnen Baugruppen und deren Speicherbelegung wieder.

CS	Adressen	Baugruppen
CS1	\$38000000..\$3BFFFFFF	FPGA INTERFACE
	\$00xx	IDREV fix
	\$02xx	SYSTEM fix
	\$20xx	IRQ
	\$40xx..\$46xx	SERIAL0
		FPGAIO[0, 2, 4]: FSR, DR, CLKR
		FPGAIO[1, 3, 5]: FSX, DX, CLKX
	\$48xx..\$4Exx	SERIAL1
		FPGAIO[6, 8, 10]: FSR, DR, CLKR
		FPGAIO[7, 9, 11]: FSX, DX, CLKX
	\$50xx..\$56xx	SERIAL2
		FPGAIO[12, 14, 16]: FSR, DR, CLKR
		FPGAIO[13, 15, 17]: FSX, DX, CLKX
	\$58xx..\$5Exx	SERIAL3
		FPGAIO[18, 20, 22]: FSR, DR, CLKR
		FPGAIO[19, 21, 23]: FSX, DX, CLKX
\$60xx..\$66xx	SERIAL4	
	FPGAIO[24, 26, 28]: FSR, DR, CLKR	
	FPGAIO[25, 27, 29]: FSX, DX, CLKX	
\$60xx..\$6Exx	SERIAL5	
	FPGAIO[30, 32, 34]: FSR, DR, CLKR	
	FPGAIO[31, 33, 35]: FSX, DX, CLKX	
\$70xx..\$76xx	SERIAL6	
	FPGAIO[36, 38, 40]: FSR, DR, CLKR	
	FPGAIO[37, 39, 41]: FSX, DX, CLKX	
\$70xx..\$7Exx	SERIAL7	
	FPGAIO[42, 44, 46]: FSR, DR, CLKR	
	FPGAIO[43, 45, 47]: FSX, DX, CLKX	

Tabelle 27: Speicherbelegung, SERC3X

3.5.3.2.1 IRQ

Dieses Register enthält IRQ-Informationen dieser Schnittstelle

ADDR	R/W	Bit	Name	Reset	ERWEITERUNG
\$20xx	R	[0]	IRQ_R0	0	Receive Interrupt SERIAL0
	R	[1]	IRQ_R1	0	Receive Interrupt SERIAL1
	R	[2]	IRQ_R2	0	Receive Interrupt SERIAL2
	R	[3]	IRQ_R3	0	Receive Interrupt SERIAL3
	R	[4]	IRQ_R4	0	Receive Interrupt SERIAL4
	R	[5]	IRQ_R5	0	Receive Interrupt SERIAL5
	R	[6]	IRQ_R6	0	Receive Interrupt SERIAL6
	R	[7]	IRQ_R7	0	Receive Interrupt SERIAL7
	R	[8]	IRQ_X0	0	Transmit Interrupt SERIAL0
	R	[9]	IRQ_X1	0	Transmit Interrupt SERIAL1
	R	[10]	IRQ_X2	0	Transmit Interrupt SERIAL2
	R	[11]	IRQ_X3	0	Transmit Interrupt SERIAL3
	R	[12]	IRQ_X4	0	Transmit Interrupt SERIAL4
	R	[13]	IRQ_X5	0	Transmit Interrupt SERIAL5
	R	[14]	IRQ_X6	0	Transmit Interrupt SERIAL6
	R	[15]	IRQ_X7	0	Transmit Interrupt SERIAL7
R	[31:16]	---	---	0	

Tabelle 28: IRQ REGISTER

3.5.3.2.2 SERIAL0 bis SERIAL7

Jede serielle Schnittstelle wird aus zwei 32 Bit Registern gebildet. Der generelle Aufbau wird hier exemplarisch für SERIAL0 aufgezeigt.

CS	Adressen	Baugruppen
CS1	\$38000000..\$3BFFFFFF	FPGA INTERFACE
	\$40xx	CONTROL
	\$42xx	STATUS
	\$44xx	FIFO
	\$46xx	SIZE

Tabelle 29: Speicherbelegung, SERIAL0

Für die anderen Schnittstellen verschieben sich die Adressen entsprechend.

CONTROL

In diesem Register wird die Funktionsweise der seriellen Schnittstelle eingestellt.

Bei einem Schreibzugriff, wird das interne Register beschrieben. Ein Lesezugriff liest den Wert des internen Registers.

ADDR	R/W	Bit	Name	Reset	
\$40xx	R/W	[0]	RUN	0	
	R	[1]	---	0	
	R/W	[2]	IRQ_X	0	
	R/W	[3]	IRQ_R	0	
	R/W	[4]	HS	0	
	R	[7:5]	---	0	
	R/W	[15:8]	TIMER	0	
	R/W	[23:16]	COUNTER	0	
	R	[31:24]	---	0	

Tabelle 30: SERIALX, CONTROL

RUN	Ist dieses Bit „HIGH“ wird die seriellen Schnittstelle freigegeben. Ein „LOW“ deaktiviert sie und versetzt das FIFO und die Handshake-Logik in den Grundzustand.
IRQ_X	Ist dieses Bit „HIGH“ wird ein Interrupt ausgelöst, wenn der Sendepuffer fast leer ist (ALMOST_EMPTY_X).
IRQ_R	Ist dieses Bit „HIGH“ wird ein Interrupt ausgelöst, wenn Daten im Empfangspuffer verfügbar sind. Hierbei werden die Statusausgänge N_EMPTY_R und ALMOST_FULL_R ausgewertet. Das ALMOST_FULL_R Flag löst hierbei sofort einen Interrupt aus, während das N_EMPTY_R-Flag, um den COUNTER-Wert, verzögert wird.
HS	Ist dieses Bit „HIGH“ wird der Handshake-Mode aktiviert. Dies entspricht im „User's Guide TMS320C3x“ der seriellen Schnittstelle „HS=1“ des „Serial Global Control Register“.
TIMER	Dieser 8 Bit-Wert definiert die Baudrate für den Sendeteil der seriellen Schnittstelle. Die Baudrate wird vom halben Systemtakt abgeleitet. Für den Systemtakt von 80MHz gilt: Baudrate = 40MHz / (TIMER + 1) TIMER = (40MHz / gewünschte Baudrate) - 1
COUNTER	Dieser 8 Bit-Wert legt fest, wie viele Takte im Empfangsteil kein Zeichen mehr empfangen werden darf, damit ein Interrupt ausgelöst wird, wenn das N_EMPTY_R-Flag aktiv ist. Diese Verzögerung ist erforderlich um bei einer Übertragung von mehreren Zeichen nicht jedes einzelne Zeichen aus dem Empfangspuffer zu lesen. Da auf der seriellen Schnittstelle immer 32 Bit-Werte übertragen werden, hat ein Wert kleiner 32 ebenfalls nicht den gewünschten Effekt.

STATUS

In diesem Register werden die Statusinformationen des FIFO angezeigt. Ein Lesezugriff liest den Status des implementierten FIFO.

ADDR	R/W	Bit	Name	Reset	
\$42xx	R	[12:0]	CNT_R	-	0 wenn RUN_R = LOW
	R	[13]	N_EMPTY_R	-	„LOW“ wenn RUN_R = LOW
	R	[14]	ALMOST_FULL_R	-	„LOW“ wenn RUN_R = LOW
	R	[15]	FULL_R	-	„LOW“ wenn RUN_R = LOW
	R	[28:16]	CNT_X	-	0 wenn RUN_X = LOW
	R	[29]	EMPTY_X	-	„LOW“ wenn RUN_X = LOW
	R	[30]	ALMOST_EMPTY_X	-	„LOW“ wenn RUN_X = LOW
	R	[31]	N_FULL_X	-	„LOW“ wenn RUN_X = LOW

Tabelle 31: SERIALX, CONTROL

CNT_R	Zeigt an wie viel Worte im Lese-FIFO stehen
N_EMPTY_R	Dieses Bit ist „LOW“, wenn der Lese-FIFO leer ist. Ist dieses Bit und das IRQ_R-Bit „HIGH“, wird ein Interrupt ausgelöst, wenn für mindestens „COUNTER Takte kein weiteres Zeichen empfangen wird.
ALMOST_FULL_R	Dieses Bit ist „HIGH“, wenn der Lese-FIFO mindestens 128 Worte enthält. Ist das Bit IRQ_R „HIGH“, wird sofort ein Interrupt ausgelöst.
FULL_R	Dieses Bit ist „HIGH“, wenn der Lese-FIFO voll ist. Ein Überlauf des FIFO ist in diesem Fall möglich.
CNT_X	Zeigt an wie viel Worte im Schreib-FIFO stehen.
EMPTY_X	Dieses Bit ist „HIGH“, wenn der Schreib-FIFO keine Daten mehr enthält.
ALMOST_EMPTY_X	Dieses Bit ist „HIGH“, wenn der Schreib-FIFO keine 16 Worte mehr enthält. Ist dieses Bit und das IRQ_X-Bit „HIGH“, wird ein Interrupt ausgelöst.
N_FULL_X	Dieses Bit ist „LOW“, wenn der Schreib-FIFO voll ist. Ein „HIGH“ zeigt an, dass der Schreib-FIFO nicht voll ist.

FIFO

Um den Mikroprozessor zu entlasten ist für jede serielle Schnittstelle ein Schreib- und ein Lese-FIFO implementiert worden. Das FIFO ist als 32 Bit Register implementiert, welcher nur mit 32 Bit-Zugriffen gelesen oder beschrieben werden darf.

Alle geschriebenen Daten werden auf der zugehörigen seriellen Schnittstelle (SERC3X) gesendet. Der Sendevorgang wird hierbei automatisch mit dem nächsten Wort fortgesetzt, solange noch Daten im Schreib-FIFO vorhanden sind. Der „Füllstand“ wird über Statusausgänge angezeigt. Diese können aus dem STATUS-Register gelesen werden.

Alle auf der zugehörigen seriellen Schnittstelle (SERC3X) empfangenen Daten werden in den Lese-FIFO geschrieben. Der „Füllstand“ wird über Statusausgänge angezeigt. Diese können aus dem STATUS-Register gelesen werden.

SIZE

In diesem Register wird die Größe (Anzahl 32 Bit Worte) des Schreib- bzw. des Lese-FIFO angezeigt.

ADDR	R/W	Bit	Name	Reset	
\$46xx	R	[15:0]	SIZE_R		Anzahl Wort Lese-FIFO
	R	[31:16]	SIZE_X	-	Anzahl Worte Schreib-FIFO

Tabelle 32: SERIALX, CONTROL

4 Steckerbelegung



Abbildung 4: Stecker

4.1 Pfostenstecker

X1				X2			
3,3V	1	2	3,3V	GND	1	2	GND
3,3V	3	4	3,3V	FPGAIO47	3	4	FPGAIO46
FPGAIO39	5	6	FPGAIO38	FPGAIO45	5	6	FPGAIO44
FPGAIO37	7	8	FPGAIO36	FPGAIO43	7	8	FPGAIO42
FPGAIO35	9	10	FPGAIO34	FPGAIO41	9	10	FPGAIO40
FPGAIO33	11	12	FPGAIO32	GND	11	12	GPIO7
FPGAIO31	13	14	FPGAIO30	GPIO6	13	14	GPIO5
FPGAIO29	15	16	FPGAIO28	GPIO4	15	16	GPIO3
FPGAIO27	17	18	FPGAIO26	GPIO2	17	18	GPIO1
FPGAIO25	19	20	FPGAIO24	GPIO0	19	20	GPIO15
FPGAIO23	21	22	FPGAIO22	GPIO14	21	22	GPIO13
FPGAIO21	23	24	FPGAIO20	GPIO12	23	24	GPIO11
FPGAIO19	25	26	FPGAIO18	GPIO10	25	26	GPIO9
FPGAIO17	27	28	FPGAIO16	GPIO8	27	28	TIMER1
FPGAIO15	29	30	FPGAIO14	TIMER0	29	30	UTCD0
FPGAIO13	31	32	FPGAIO12	URXD0	31	32	UTXD1
FPGAIO11	33	34	FPGAIO10	URXD1	33	34	/UCTS0
FPGAIO9	35	36	FPGAIO8	/URTS0	35	36	/UCTS1
FPGAIO7	37	38	FPGAIO6	/URTS1	37	38	CANRX
FPGAIO5	39	40	FPGAIO4	CANTX	39	40	GND
FPGAIO3	41	42	FPGAIO2	/RSTIN	41	42	/RSTOUT
FPGAIO1	43	44	FPGAIO0	GND	43	44	EXTBAT
GND	45	46	GND	GND	45	46	QSPID0
GND	47	48	ETH TX-	QSPIDI	47	48	QSPICLK
GND	49	50	ETH TX+	QSPICS0	49	50	QSPICS1
/LEDRX	51	52	TXCP	QSPICS2	51	52	/QSPIINT
/LEDLNK	53	54	ETH RX-	QSPIZB0	53	54	QSPIZB1
/LED100	55	56	ETH RX+	QSPIZB2	55	56	QSPIZB3

Tabelle 33: Pin Belegung Pfostenstecker

4.2 BDM, JTAG

Die CPU verfügt über eine IEEE 1149.1 Schnittstelle. Leider können andere Bauteile - ohne zusätzliche Hardware - nicht mit der CPU in einer JTAG Kette verbunden werden, weil diese Schnittstelle zusammen mit dem BDM (Background Debug Mode) genutzt wird.

Beide Schnittstellen sind über einen „Edge-Connector“ mit SMD Anschlüssen (0,8 mm) am Platinenrand herausgeführt. Als Gegenstecker kann unter anderem die Samtec Buchsenleisten der Serie HSEC8 eingesetzt werden.

Lötseite, LS				Bauteilseite, BS					
				Anschlüsse					
J T A G	VCC _{TARGET} ³	3,3V		1	2	3,3V	³ VCC _{TARGET}	J T A G	
	Logik Ausgang	TDO	O	3	4	I	TDI		Logik Eingang
	Logik Eingang	TMS	I	5	6	I	TCK		Logik Eingang
	Masse	GND		7	8		GND		Masse
	NC ²			9	10		² NC		
	NC ²			11	12		² NC		
	NC ²			13	14		² NC		
	NC ²			15	16		² NC		
	NC ²			17	18		² NC		
D E B U G / B D M	Masse	GND		19	20		GND	Masse	D E B U G / B D M
	NC ²			21	22	I	/RSTIN	MCF5281	
	MCF5281	CLKOUT	O	23	24	I	/TA	MCF5281	
	Masse	GND		25	26		GND	Masse	
	MCF5281	/BKPT	I	27	28	I	DSCLK	MCF5281	
	MCF5281	DSI	I	29	30	O	DSO	MCF5281	
	Prozessor Status ¹	PST0	O	31	32	O	PST3	¹ Prozessor Status	
	Prozessor Status ¹	PST2	O	33	34	O	PST1	¹ Prozessor Status	
	Debug Daten ¹	DDATA0	O	35	36	O	DDATA2	¹ Debug Daten	
	Debug Daten ¹	DDATA1	O	37	38	O	DDATA3	¹ Debug Daten	
VCC _{TARGET} ³	3,3V	I	39	40		3,3V	³ VCC _{TARGET}		

Tabelle 34: JTAG/BDM Signale

- ¹ Diese Signale sind bei Motorola für den BDM definiert, werden vom Abatron BDI aber nicht unterstützt. Am parallelen BDM Stecker (Freescale, Programmieren internes FLASH) sind diese jedoch angeschlossen.
- ² Diese Anschlüsse sind nicht belegt.
- ³ Diese Signale werden vom Modul gespeist und können von einem Aufsatz (JTAG/BDM) als Versorgungsspannung für die Ausgangstreiber der Signale benutzt werden.

5 Auslieferungszustand

Der eWebSrv wird mit dem FPGA IO48 und der Extension für die eWebTest Trägerkarte ausgeliefert. Selbstverständlich kann auch eine OEM-Version mit kundenspezifischem FPGA und Konfiguration geliefert werden.

6 Inbetriebnahme

Der **eWebSrv** ist, wie oben bereits beschrieben, ein Gerät, welches über das Netzwerk betrieben wird. Daher ist auch die erste Inbetriebnahme und Konfiguration so implementiert worden, dass diese Arbeiten einfach und universell über das Netzwerk erfolgen können.

Die Beschreibung dieser Inbetriebnahme, sowie die Erläuterung der erforderlichen Netzwerkinfrastruktur befindet sich in dem separaten Handbuch „Inbetriebnahme“.

7 Konfiguration, Web Management Interface

Durch die Möglichkeit, den **eWebSrv** mit wenigen Mausklicks zu konfigurieren, wird das Gerät schnell für jede Anwendung angepasst und eingerichtet. Die implementierte Anwenderoberfläche wird mit einem Standard-Browser dargestellt. Dadurch ist es möglich, den **eWebSrv** von jedem netzwerkfähigen Rechner aus zu bedienen.

Die Beschreibung der Konfiguration mit dem „Web Management Interface“ befindet sich in dem separaten Handbuch „Benutzer“.