Realzeitplattform Mikrocontroller + CPLD

Stand: 11. 11. 2013

Es ist eine Demonstrationsplattform zu schaffen, die Untersuchungen, Vorführungen, Übungen usw. zu folgenden Problemkreisen ermöglicht:

1. Das Zusammenwirken von zwei Mikrocontrollern, wobei der eine harte Realzeitaufgaben und der andere die Kommunikation mit weiteren Systemen übernimmt, beispielsweise mit einem Personalcomputer. Beide Abläufe sind asynchron zueinander. Die Kommunikationsvorgänge dürfen das Realzeitverhalten nicht beeinflussen. Das Zeitverhalten der Realzeitmaschine muß immer deterministisch bleiben; wenn nötig, bis auf den Takt genau. Deshalb dürfen in der Realzeitmaschine weiter Interrupts wirksam werden noch darf sie in Wartezustände unvorhersagbarer Dauer gelangen. Beide Maschinen sollen auf einfache Weise miteinander gekoppelt werden. Absolutes Leistungsvermögen ist nicht entscheidend; es geht ums Prinzip. Wir wollen versuchen, ohne den Dual Port RAM auszukommen, der herkömmlicherweise in solchen Konfigurationen üblich ist.

2. Das Zusammenwirken eines Mikrocontrollers mit einem CPLD. Solche Lösungen werden gelegentlich von CPLD-Herstellern propagiert. Es sind zwei Schaltkreise, aber beide sind kostengünstig. Kopplung und Funktionstrennung sind überschaubar. Der Mikrocontroller wird in Assembler oder (beispielsweise) C programmiert, die Funktionen im CPLD beispielsweise mit Verilog oder VHDL. Die Kommunikation zwischen beiden Einrichtungen kann leicht beobachtet werden (Logikanalysator). Es ist im Grunde eine Vorstufe zur echten Einchiplösung (FPGA mit Prozessorkern(en) und anwendungsspezifischen Einzweckschaltungen).

Der CPLD-Schaltkreis soll auf einer Aufsteckplatine untergebracht werden, um unterschiedliche CPLDs zum Einsatz bringen zu können.

Die Anordnung soll mit 3,3 V betrieben werden können. Debugging-Vorkehrungen sind zu untersuchen (Testpunkte, Schnittstellen, Ports zum Anschließen von Bediengeräten).

Kommunikation mit übergeordneten Systemen über die serielle Schnittstelle.

Vorläufige Entscheidung:
Zwei Platinen (Mikrocontrollerplatine, CPLD-Platine). Formfaktor für kleines Teko-Gehäuse (ca. 92 * 156 mm; wie Port Sniffer usw.). Bandkabelverbindung. Ggf. Vorkehrungen zum Übereinanderstapeln beider Platinen. (Piggyback-Prinzip.) CPLD-Platine oben.

CPLD-Platine ohne Programmer.

Zuordnung der Ports zu den CPLD-Pins:

Bit	Port A	Port B	Port C	Port D	Port E	Port F	Port G	Port H	Port I
0	1	2	3	4	5	6	7	11	12 (GCK3)
1	71	72	75	79	80	81	82	83	10 (GCK2)
2	14	15	17	18	19	20	21	23	76 (GTS1)
3	57	58	61	62	63	65	66	67	9 (GCK1)
4	32	33	34	35	36	37	39	40	74 (GSR)
5	45	46	47	48	50	51	52	53	
6	13	24	25	26	41	43	44	31	
7	84	68	69	70	54	55	56	77 (GTS2)	

Portübersicht
Aufbau der folgenden Tabellen:

1. Bitindex im Port.
2. Kontakt des CPLD.
3. Steckverbinderkontakt.

Port A:

7	6	5	4	3	2	1	0
84	13	45	32	57	14	71	1

Port B:

7	6	5	4	3	2	1	0
68	24	46	33	58	15	72	2

Port C:

7	6	5	4	3	2	1	0
69	25	47	34	61	17	75	3

Port D:

7	6	5	4	3	2	1	0
70	26	48	35	62	18	79	4

Port E:

7	6	5	4	3	2	1	0
54	41	50	36	63	19	80	5

Port F:

7	6	5	4	3	2	1	0
55	43	51	37	65	20	81	6

Port G:

7	6	5	4	3	2	1	0
56	44	52	39	66	21	82	7

Port H:

7	6	5	4	3	2	1	0
77 = GTS2	31	53	40	67	23	83	11

Port I:

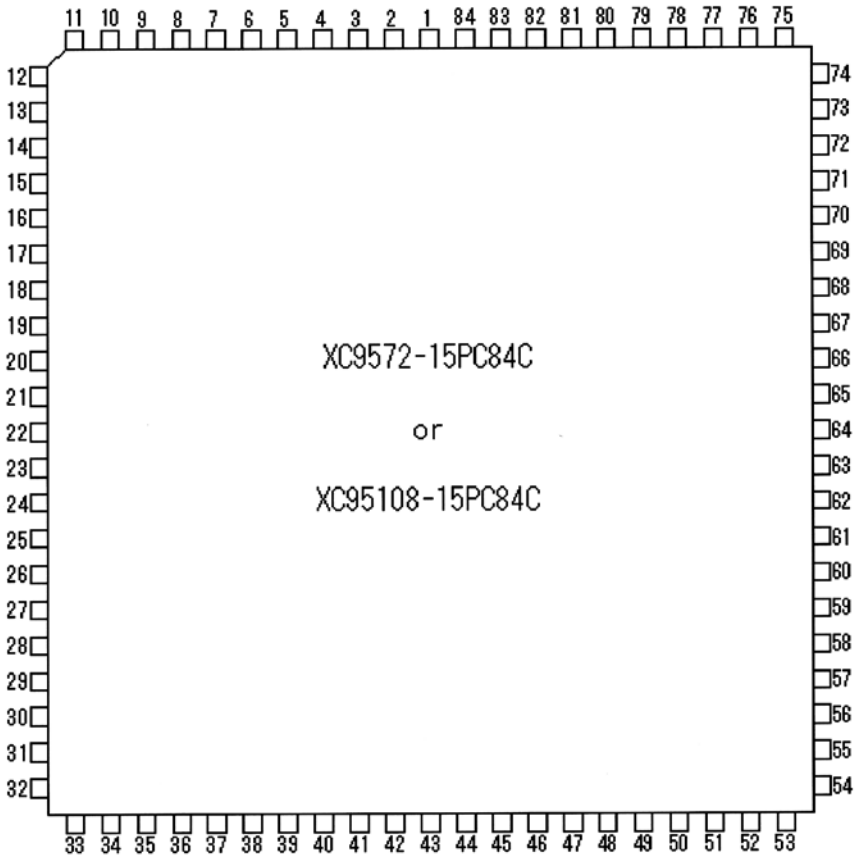
7	6	5	4	3	2	1	0
–	–	–	74	9	76	10	12
–	–	–	GSR	GCK1	GTS1	GCK2	GCK3

Function Block	Macrocell	PC84	Port	Notes
1	1	–		
1	2	1	**A0**	
1	3	2	**B0**	
1	4	–		
1	5	3	**C0**	
1	6	4	**D0**	
1	7	–		
1	8	5	**E0**	
1	9	6	**F0**	
1	10	–		
1	11	7	**G0**	
1	12	9	**T1**	[1]
1	13	–		
1	14	10	**T2**	[1]
1	15	11	**H0**	
1	16	12	**T3**	[1]
1	17	13	**S0**	
1	18	–		
2	1	–		
2	2	71	**A1**	
2	3	72	**B1**	
2	4	–		
2	5	74	**T0**	[1]
2	6	75	**C1**	
2	7	–		
2	8	76	**T4**	[1]
2	9	77	**T5**	[1]
2	10	–		
2	11	79	**D1**	
2	12	80	**E1**	
2	13	–		
2	14	81	**F1**	
2	15	82	**G1**	
2	16	83	**H1**	
2	17	84	**S1**	
2	18	–		

Function Block	Macrocell	PC84	Port
3	1	–	
3	2	14	**A2**
3	3	15	**B2**
3	4	–	
3	5	17	**C2**
3	6	18	**D2**
3	7	–	
3	8	19	**E2**
3	9	20	**F2**
3	10	–	
3	11	21	**G2**
3	12	23	**H2**
3	13	–	
3	14	24	**S2**
3	15	25	**A6**
3	16	26	**B6**
3	17	31	**C6**
3	18	–	
4	1	–	
4	2	57	**A3**
4	3	58	**B3**
4	4	–	
4	5	61	**C3**
4	6	62	**D3**
4	7	–	
4	8	63	**E3**
4	9	65	**F3**
4	10	–	
4	11	66	**G3**
4	12	67	**H3**
4	13	–	
4	14	68	**S3**
4	15	69	**D6**
4	16	–	
4	17	70	**E6**
4	18	–	

Function Block	Macrocell	PC84	Port
5	1	–	
5	2	32	**A4**
5	3	33	**B4**
5	4	–	
5	5	34	**C4**
5	6	35	**D4**
5	7	–	
5	8	36	**E4**
5	9	37	**F4**
5	10	–	
5	11	39	**G4**
5	12	40	**H4**
5	13	–	
5	14	41	**F6**
5	15	43	**A7**
5	16	–	
5	17	44	**B7**
5	18	–	
6	1	–	
6	2	45	**A5**
6	3	46	**B5**
6	4	–	
6	5	47	**C5**
6	6	48	**D5**
6	7	–	
6	8	50	**E5**
6	9	51	**F5**
6	10	–	
6	11	52	**H5**
6	12	53	**C7**
6	13	–	
6	14	54	**D7**
6	15	55	**E7**
6	16	–	
6	17	56	**F7**
6	18	–	

Pin Type	PC84
I/O/GCK1	9
I/O/GCK2	10
I/O/GCK3	12
I/O/GTS1	76
I/O/GTS2	77
I/O/GSR	74
TCK	30
TDI	28
TDO	59
TMS	29
V_{CCINT} 5 V	38,73,78
V_{CCIO} 3.3 V/5 V	22,64
GND	8,16,27,42,49,60


11 10 9 8 7 6 5 4 3 2 1 84 83 82 81 80 79 78 77 76 75
12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32
74 73 72 71 70 69 68 67 66 65 64 63 62 61 60 59 58 57 56 55 54
33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53
XC9572-15PC84C
or
XC95108-15PC84C

Vermischte Notizen:

Ein besonderer Formfaktor wird nicht gefordert.

Ein Interface, das ausschließlich mit Software gesteuert wird, kann für beide Einrichtungen (Kommunikationsprozessor und CPLD) gemeinsam genutzt werden.

Ziel: schnelle, leicht programmierbare Statusabfrage.

Unabhängige Signale sind mit Bittests oder durch Einlesen direkt abfragbar.

Busprinzip erfordert hingegen Auswahl:
OUT sense_select
NOP ; Warten wegen Synchronisation
IN sense_data

Das CPLD kann auf ein Auswahlsignal sofort reagieren, nicht aber ein Mikrocontroller. Busaufschaltung nur über Software möglich. Damit braucht man eine Rückmeldung, und die Wartezeit wird unbestimmt. Der Kommnunikationsprozessor kann Interrupts verarbeiten (z. B. von der seriellen Schnittstelle), dadurch wird sein Realzeitverhalten undefiniert.

Der Realzeitprozessor darf keine undefinierten Wartezustände zu sehen bekommen. Wir brauchen im Grunde ein Verhalten wie ein Dual Port RAM.

Einen echten DP RAM setzen.
Einen kleinen DP RAM mit CPLD nachbauen. Lohnt womöglich nicht. 34 Zellen ergeben 4 Bytes, 72 Zellen 8 Bytes. (Dars reicht aber, um typische Parameter zu übergeben, beispielsweise für die Pulsweitenmodulation.)
Oder FIFOs. Wir brauchen dann nur den Datenweg, keine Adressen.
Verhalten mit Software emulieren. Das geht aber nur mit Handshaking und wird langsam.
RAM setzen mit gemeinsamem Busanschluß. Damit ein Dual-Port-Verhalten nachbilden (Pufferanwendung mit vorgeschalteter Signalisierung, wo es nicht vorkommt, daß beide Einrichtungen gleichzeitig zugreifen). Wenn es nur ein Puffer ist, ginge es aber auch mit FIFOs (Kostenfrage). FIFO spart Adreßbus.

CPLD-Schnittstelle:
8 Daten (bidirektional)
4 Adresse/Kommando
REQUEST
REPLY
Kommandocode bestimmt die Übertragungsrichtung
Ein Kommando SENSE schaltet Zustandsbits auf den Bus auf.

Kommunikationsfälle Realzeitprozessor mit Kommunikationsprozessor

Die Kommunikation muß in den allgemeinen Schleifenumlauf eingebunden werden.

Wie langer darf der Umlauf dauern?
Wenn die Zeit kurz ist und wenn sie konstant bleiben soll, kann man die Kommunikation ohnehin nur stückweise erledigen, beispielsweise bei jedem Umlauf nur ein einziges Byte übertragen. Ein Programmschlenker, um z. B. einige kBytes am Stück zu transportieren, ist dann nicht möglich.

Den Funktionsablauf nur dann ändern, nachdem alle Parameter übertragen worden sind. Alle Funktionen, bei denen der Jitter der Sofware zu groß ist, müssen in Hardware erledigt werden. Dort sind ggf. Abbildungsregister vorzusehen.

Zwei identische Schnittstellen zu je 14 Signalen.
Realzeitmaschine hat 4 Signale frei.
Kommunikationsmaschine hat 18 Signale frei. Davon gehen 2 für die serielle schnittstelle nach außen ab. Dazu noch 5 für SPI (wie Universalgerät usw.). Verbleiben 11 Signale.
2. serielle Schnittstelle zur Realzeitmaschine (Debugging usw.)?
1 Aktivitäts-LED
1 freier Port, beispielsweise für die Einheitsbedientafel.

Die 4 Signale der Realzeitmaschine: 2 serielle Schnittstelle, 2 Reserve zur CPLD.

Meßpunkte nicht erforderlich. Bei Bedarf Port Sniffer zwischenschalten.