

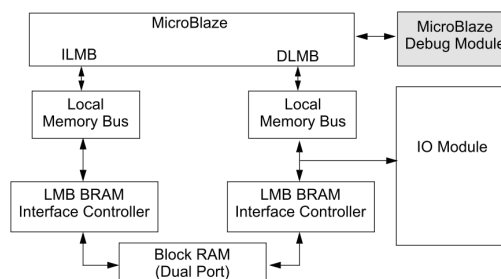


Microblaze MCS

© 2004 Xilinx, Inc. All Rights Reserved

Microblaze Micro Controller

- Tulajdonságok
- MicroBlaze processzor
- LMB BRAM memória
- MicroBlaze Debug Module (MDM)
- LMB buszra illesztett IO modulok
 - Külső IO Busz
 - Megszakítás vezérlő
 - UART
 - Fixed Interval Timers (FIT)
 - Prog.Interval Timers (PIT)
 - General Purpose Input
 - General Purpose Output



Konfiguráció 1

MicroBlaze MCS

Component Name: MBMCS

Micro Controller System

Instance Hierarchical Design Name: mcs_0

Input Clock Frequency (MHz): 16

Memory Size: 16KB

Enable ID Bus: ☒

Enable Debug Support: ☒

Enable MicroBlaze Trace Bus: ☐

PlanAhead & Project Navigator Information

Software Development Information

After generating the core, there are a few steps necessary in PlanAhead or Project Navigator, mainly to support software development.

- Before implementing the design, information about the generated BMM file describing the memory of the MicroBlaze MCS core must be included in the tool. The script `microblaze_mcs_setup.tcl` is available to do this, using source `ip-directory/microblaze_mcs_setup.tcl` in the Td console.
- Before generating the bitstream, tool options should be updated to include the software program for the MicroBlaze MCS core. This can be achieved by invoking `microblaze_mcs_data2mem` in the Td Console, with the software program ELF file as parameter.

See the datasheet for more information and examples, including how to handle more than one MicroBlaze MCS core in a project.

Generate Cancel Help

Hardware Design - 3 - 4 © 2004 Xilinx, Inc. All Rights Reserved Kizárolag oktatási célra

Konfiguráció 2

MicroBlaze MCS

Component Name: MBMCS

Micro Controller System

Instance Hierarchical Design Name: mcs_0

Input Clock Frequency (MHz): 16

Memory Size: 16KB

Enable ID Bus: ☒

Enable Debug Support: ☒

Enable MicroBlaze Trace Bus: ☐

PlanAhead & Project Navigator Information

Software Development Information

After generating the core, there are a few steps necessary in PlanAhead or Project Navigator, mainly to support software development.

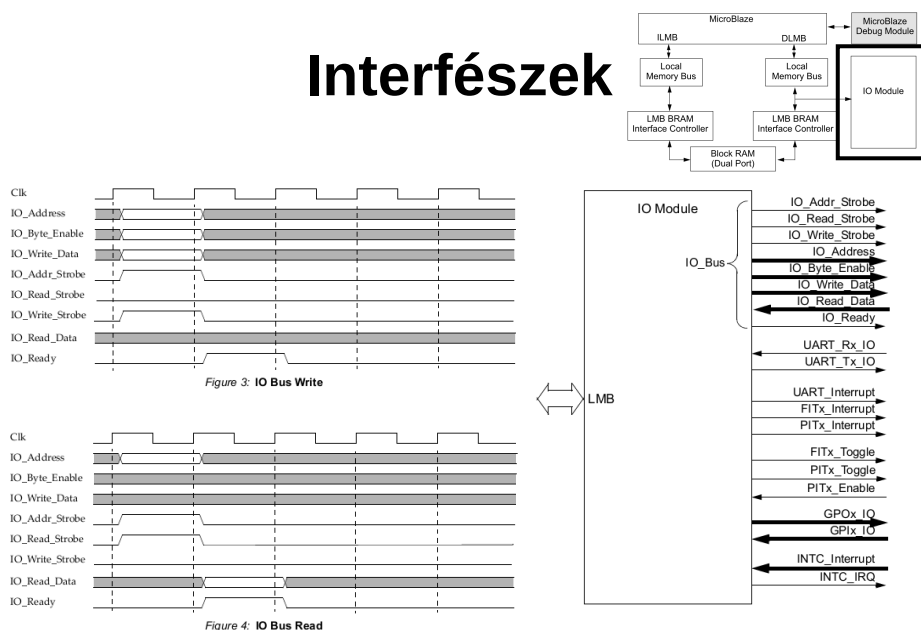
- Before implementing the design, information about the generated BMM file describing the memory of the MicroBlaze MCS core must be included in the tool. The script `microblaze_mcs_setup.tcl` is available to do this, using source `ip-directory/microblaze_mcs_setup.tcl` in the Td console.
- Before generating the bitstream, tool options should be updated to include the software program for the MicroBlaze MCS core. This can be achieved by invoking `microblaze_mcs_data2mem` in the Td Console, with the software program ELF file as parameter.

See the datasheet for more information and examples, including how to handle more than one MicroBlaze MCS core in a project.

Generate Cancel Help

Hardware Design - 3 - 5 © 2004 Xilinx, Inc. All Rights Reserved Kizárolag oktatási célra

Interfészek

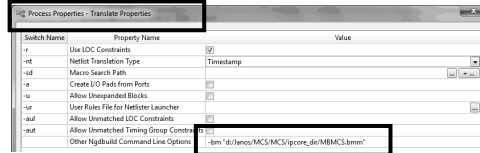
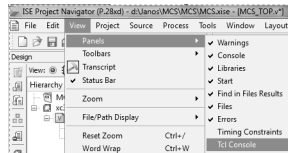


Címtartomány

Address (hex)	Name	Access Type	Description
0x0-C_MEMSIZE-1	Local Memory	RW	Local Memory for MicroBlaze software
C_MEMSIZE-0x7FFFFFFF	Reserved		
0x8000000	UART_RX	R	UART Receive Data Register
0x8000004	UART_TX	W	UART Transmit Data Register
0x8000008	UART_STATUS	R	UART Status Register
0x800000C	Reserved		
0x8000010	GPO1	W	General Purpose Output 1 Register
0x8000014	GPO2	W	General Purpose Output 2 Register
0x8000018	GPO3	W	General Purpose Output 3 Register
0x800001C	GPO4	W	General Purpose Output 4 Register
0x8000020	GPI1	R	General Purpose Input 1 Register
0x8000024	GPI2	R	General Purpose Input 2 Register
0x8000028	GPI3	R	General Purpose Input 3 Register
0x800002C	GPI4	R	General Purpose Input 4 Register
0x800007C	Reserved		
0x8000080-0xBF FFFF	Reserved		
0xC000000-0xFFFF FFFF	IO Bus	RW	Mapped to IO Bus address output IO_Address



Implementáció



source ipcore_dir/microblaze_mcs_setup.tcl

Példányosítás

```
1 timescale 1ns / 1ps
2
3 module MCS_TOP
4     input clk16,
5     input reset,
6     input uart_tx,
7     output uart_tx,
8     output uart_rx,
9     input [7:0] sw,
10    output [7:0] led
11
12
13
14
15
16 HBMC2 mcs_0 (
17     .clk(clk16), // input clk
18     .Reset(reset), // input Reset
19
20     .UART_Rx(uart_rx), // input UART_Rx
21     .UART_Tx(uart_tx), // output UART_Tx
22     .GPIO_Led0[7:0] (sw), // output [7:0] GPIO
23     .GPIO_Led1[7:0] (sw), // output [7:0] GPIO
24     .GPIO_Interrupt[0] (sw), // output GPIO_Interrupt
25
26
27 endmodule
```

Erőforrás kihasználás

Parameter Values (other parameters at default value)													Device Resources		
C_USE_UART_RX	C_USE_UART_TX	C_INTRC_USE_EXT_INTR	C_INTRC_INTR_SIZE	C_USE_FIT1	C_FIT1_No_CLOCKS	C_USE_PIT1	C_PIT1_SIZE	C_USE_GPH1	C_GPH1_SIZE	C_USE_GPO1	C_GPO1_SIZE	C_USE_IO_BUS	C_DEBUG_ENABLE	LUTs	Flip-Flops
1	1	0	0	0	0	0	0	0	0	0	0	0	0	716	299
1	1	1	5	0	0	0	0	0	0	0	0	0	0	733	330
1	1	1	5	1	65000	0	0	0	0	0	0	0	0	740	342
1	1	1	5	1	65000	1	32	0	0	0	0	0	0	783	434
1	1	1	5	1	65000	1	32	1	32	0	0	0	0	804	466
1	1	1	5	1	65000	1	32	1	32	1	32	0	0	805	498
1	1	1	5	1	65000	1	32	1	32	1	32	1	0	820	602
1	1	1	5	1	65000	1	32	1	32	1	32	1	1	1022	959

Hardware Design - 3 - 8

© 2004 Xilinx, Inc. All Rights Reserved

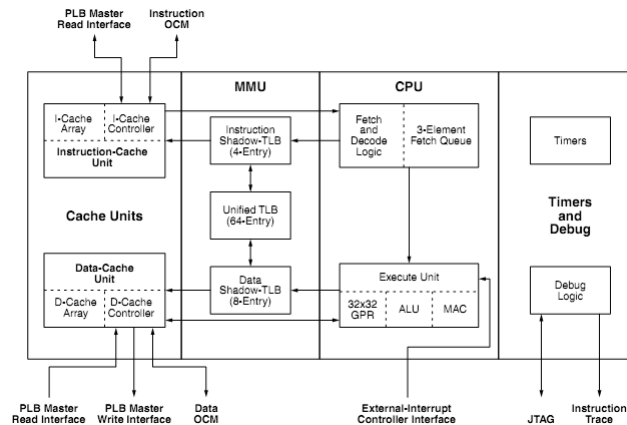
Kizárólag oktatási célra



PowerPC

© 2004 Xilinx, Inc. All Rights Reserved

A PowerPC processzor



Megjegyzés: Az OCM busz a valóságban nincs kapcsolatban a cache vezérlővel



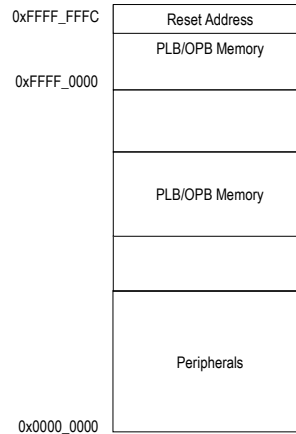
A PowerPC processzor

- A PowerPC™ megvalósít egy 32 bites beágyazott környezet architektúra
- Beágyazott rendszer alkalmazások támogatása
 - Rugalmas memória menedzsment
 - Szorzás és összegzés utasítás a számításigényes alkalmazásokhoz
 - Javított hibakeresési képességek
 - 64 bites időalap
 - Programozható (PIT), és fix (FIT) intervallum időzítők, és "watchdog" időzítők
- Teljesítménynövelő tulajdonságok
 - Statikus elágazás jóslás
 - 5 fokozatú futószalag
 - Hardver szorzás/osztás a gyors egész aritmetikához
 - Javított karakterlánc és többszörös szó kezelés
 - Minimalizált megszakítás késleltetés



PowerPC

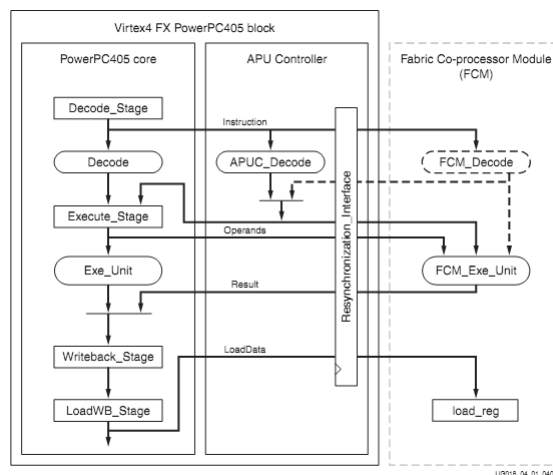
- Memória és perifériák
 - A PPC405 32 bites címzést használ
- Speciális címek
 - Minden PowerPC™ rendszerben a RESET vektor címe 0xFFFFFFF, tehát itt kell lennie az indító szektornak
 - A minimális programmemória a 0xFFFF0000 to 0xFFFFFFFF címek közötti folytonos címtartományt tölti ki.
 - Ha van megszakítás kezelő, akkor a megszakításvektornak egy 64k blokkhatáron kell lennie



PowerPC™



Fabric Co-processor module



U2018_04_01_040204



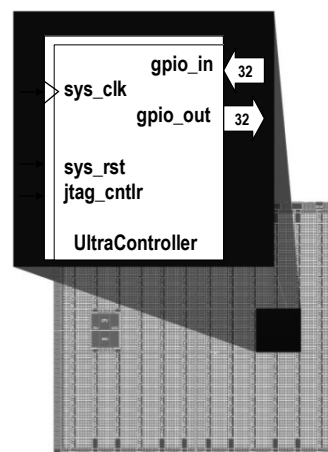


Ultra Controller

© 2004 Xilinx, Inc. All Rights Reserved

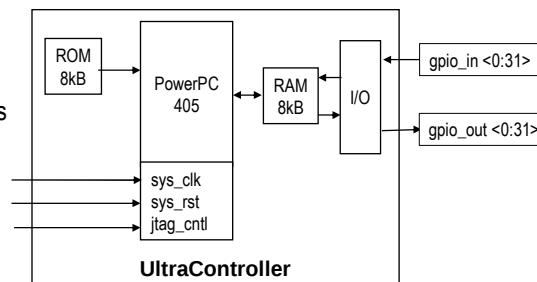
UltraController

- UltraController™ fő jellemzők
 - Teljesen önálló PowerPC™ rendsze
 - 32 általános célú be és kimenet
 - Ultra alacsony fogyasztás— 0.9 mW / MHz
- Könnyű használat
 - "C" kód — Referencia tervek elérhetők
 - **Integrálható az ISE szabványos tervezési folyamatába**
 - Teljes fejlesztési és hibakeresési támogatás
- Az UltraController™ rugalmas alkalmazási lehetőségeket biztosít
 - Összetett felhasználói interfészek (GUI vagy LCD kijelző)
 - Bonyolult adat műveletek, matematikai operációk
 - Rendszer monitorozás, statisztikák gyűjtése
 - Komplex algoritmusok



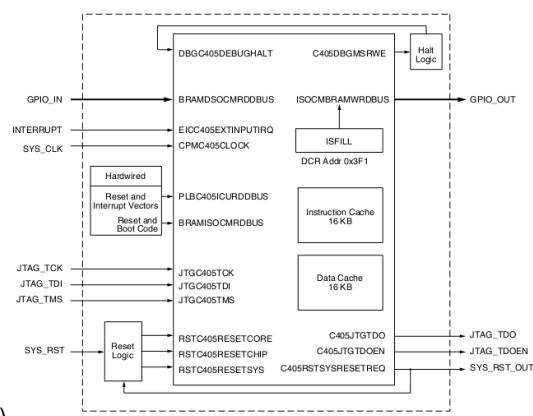
Az UltraController I

- A PowerPC processzoron alapuló általános célú vezérlő
 - Növekvő számú alkalmazások
 - 8 vagy 16 BRAM-ot használ + 49 logikai cellát
 - 8-kB utasítás, 8-kB adat
 - 16-kB utasítás, 16-kB adat
 - JTAG fejlesztési támogatás



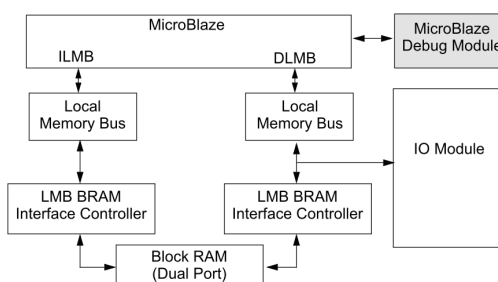
Az UltraController II.

- Magas CPU órajel
 - Max. 450 MHz Virtex-4 FX
 - Max 400 MHz Virtex-II Pro
- **Programkód a Cache-ben**
 - 16 KB I-side
 - 16 KB D-side
- **BRAM-ot nem használ**
- 32-bit output
- 32-bit input
- Programmable interval timer (PIT)
- Fixed interval timer (FIT)
- Watchdog timer (WDT)
- Egyetlen megszakítás vonal (EXT)

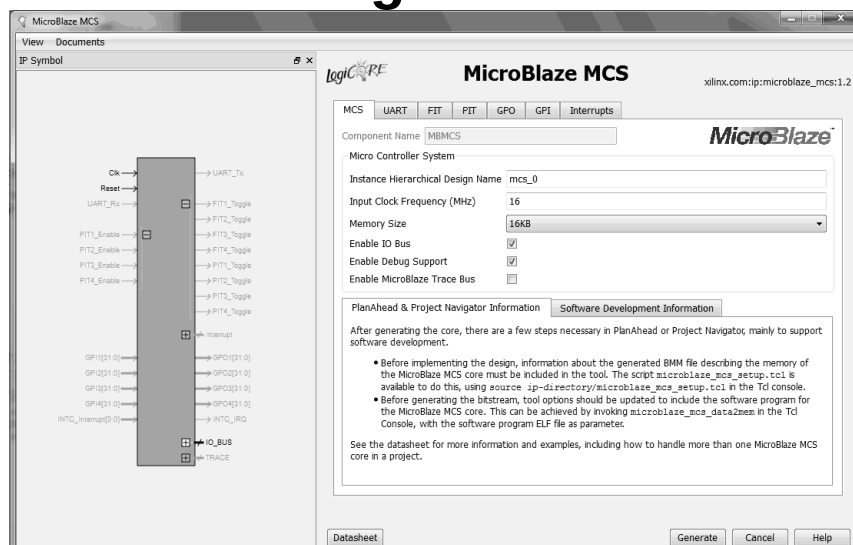


Microblaze Micro Controller

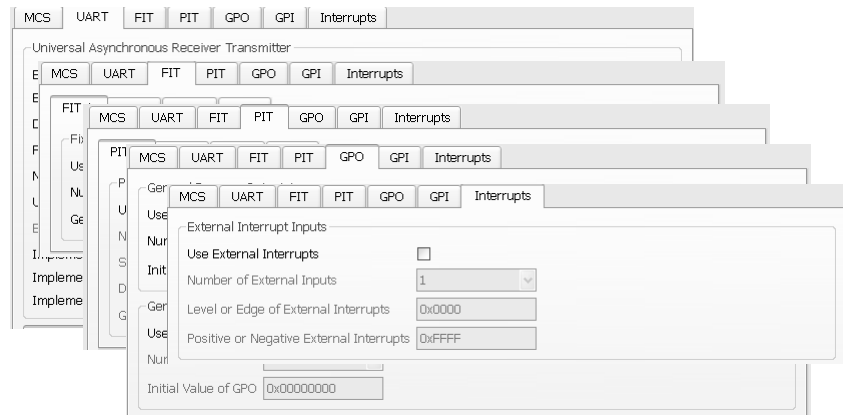
- Tulajdonságok
- MicroBlaze processzor
- LMB BRAM memória
- MicroBlaze Debug Module (MDM)
- LMB buszra illesztett IO modulok
 - Külső IO Busz
 - Megszakítás vezérlő
 - UART
 - Fixed Interval Timers (FIT)
 - Prog.Interval Timers (PIT)
 - General Purpose Input
 - General Purpose Output



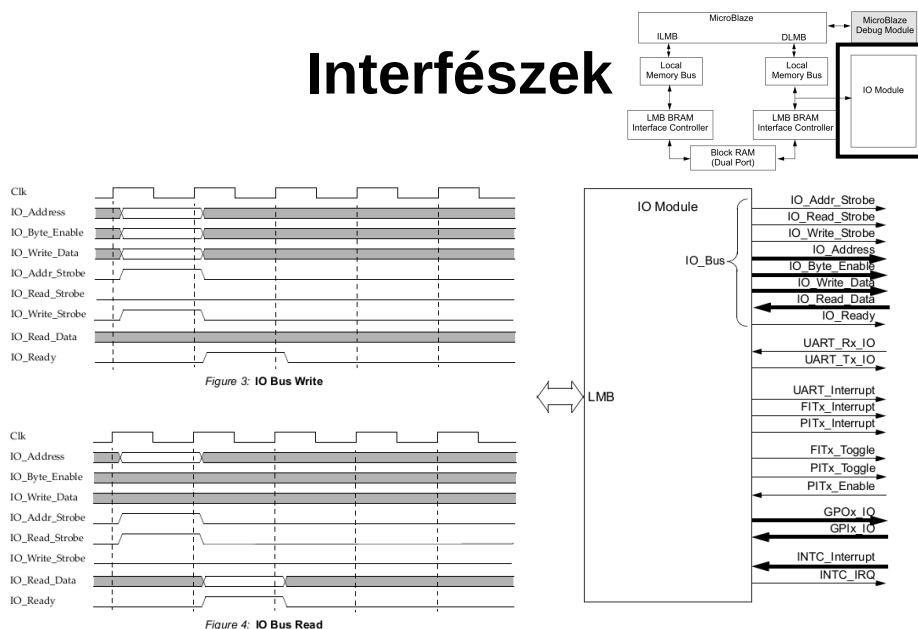
Konfiguráció 1



Konfiguráció 2



Interfészek

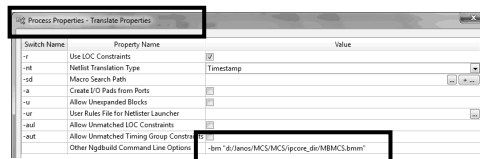
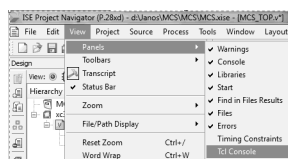


Címtartomány

Address (hex)	Name	Access Type	Description
0x0-C_MEMSIZE-1	Local Memory	RW	Local Memory for MicroBlaze software
C_MEMSIZE-0xFFFFFFFF	Reserved		
0x8000000	UART_RX	R	UART Receive Data Register
0x8000004	UART_TX	W	UART Transmit Data Register
0x8000008	UART_STATUS	R	UART Status Register
0x800000C	Reserved		
0x8000010	GPO1	W	General Purpose Output 1 Register
0x8000014	GPO2	W	General Purpose Output 2 Register
0x8000018	GPO3	W	General Purpose Output 3 Register
0x800001C	GPO4	W	General Purpose Output 4 Register
0x8000020	GPI1	R	General Purpose Input 1 Register
0x8000024	GPI2	R	General Purpose Input 2 Register
0x8000028	GPI3	R	General Purpose Input 3 Register
0x800002C	GPI4	R	General Purpose Input 4 Register
0x800007C	Reserved		
0x8000080-0xBFFFFFFF	Reserved		
0xC0000000-0xFFFFFFFF	IO Bus	RW	Mapped to IO Bus address output IO_Address



Implementáció



source ipcore_dir/microblaze_mcs_setup.tcl

Példányosítás

```

1 timescale 1ns / 1ps
2
3 module MCS_TOP (
4     input clk16,
5     input reset,
6
7     input uart_tx,
8     output uart_rx,
9
10    input [7:0] pw,
11    output [7:0] led
12)
13
14
15
16 MBMC2 mcs_0 (
17     .clk(clk16), // input Clk
18     .reset(reset), // input Reset
19
20     .UART_Rx(uart_rx), // input UART Rx
21     .UART_Tx(uart_tx), // output UART Tx
22     .GPI0(led), // output [7 : 0] GPI0
23     .GPI1(pw), // input [7 : 0] GPI1
24     .GPI2_interrupt() // output GPI2_interrupt
25 );
26
27 endmodule
28

```

Erőforrás kihasználás

Parameter Values (other parameters at default value)													Device Resources		
C_USE_UART_RX	C_USE_UART_TX	C_USE_UART_EXT_INTR	C_INTRC_INTR_SIZE	C_INTRC_FIT1	C_FIT1_No_CLOCKS	C_USE_PIT1	C_PIT1_SIZE	C_USE_GPH1	C_GPH1_SIZE	C_USE_GPO1	C_GPO1_SIZE	C_USE_IO_BUS	C_DEBUG_ENABLE	LUTs	Flip-Flops
1	1	0	0	0	0	0	0	0	0	0	0	0	0	716	299
1	1	1	5	0	0	0	0	0	0	0	0	0	0	733	330
1	1	1	5	1	65000	0	0	0	0	0	0	0	0	740	342
1	1	1	5	1	65000	1	32	0	0	0	0	0	0	783	434
1	1	1	5	1	65000	1	32	1	32	0	0	0	0	804	466
1	1	1	5	1	65000	1	32	1	32	1	32	0	0	805	498
1	1	1	5	1	65000	1	32	1	32	1	32	1	0	820	602
1	1	1	5	1	65000	1	32	1	32	1	32	1	1	1022	959





Saját modul illesztése (BBD használata)

© 2004 Xilinx, Inc. All Rights Reserved

MPD fájl

MPD file created automatically for design OPB_SEMAPHORE

BEGIN opb_pwm, IPTYPE=PERIPHERAL

Parameter list for the generics

PARAMETER C_OPB_AWIDTH = 32, DT = integer
PARAMETER C_OPB_DWIDTH = 32, DT = integer
PARAMETER C_BASEADDR = 0xFFFF8000, DT = std_logic_vector
PARAMETER C_HIGHADDR = 0xFFFF80FF, DT = std_logic_vector
PARAMETER C_NO_CHANNELS = 4, DT = integer
PARAMETER C_MAX_RESOLUTION = 16, DT = integer

OPTION SIM_MODELS = BEHAVIORAL : STRUCTURAL

BUS_INTERFACE BUS=SOPB, BUS_STD=OPB, BUS_TYPE=SLAVE

A paraméterek felülírják a VHDL generic-et

```
entity OPB_PWM is
generic (
  C_OPB_AWIDTH   : integer           := 32;
  C_OPB_DWIDTH   : integer           := 32;
  C_BASEADDR     : std_logic_vector(0 to 31) := "FFFFFFA000";
  C_HIGHADDR     : std_logic_vector    := "FFFFFFA0FF";
  C_NO_CHANNELS  : integer range 0 to 15 := 4;
  C_MAX_RESOLUTION : integer range 4 to 32 := 16
);
```



MPD fájl

```
## Port list for the signals
## Global signals
PORT OPB_Clk = "", DIR = in, SIGIS=CLK, BUS=SOPB
PORT OPB_Rst = OPB_Rst, DIR = in, BUS=SOPB
## OPB signals
PORT OPB_ABus = OPB_ABus, DIR = in, VEC = [0:31], BUS=SOPB
PORT OPB_BE = OPB_BE, DIR = in, VEC = [0:3], BUS=SOPB
PORT OPB_RNW = OPB_RNW, DIR = in, BUS=SOPB
PORT OPB_select = OPB_select, DIR = in, BUS=SOPB
PORT OPB_seqAddr = OPB_seqAddr, DIR = in, BUS=SOPB
PORT OPB_DBus = OPB_DBus, DIR = in, VEC = [0:31], BUS=SOPB

PORT PWM_DBus = SI_DBus, DIR = out, VEC = [0:31], BUS=SOPB
PORT PWM_errAck = SI_errAck, DIR = out, BUS=SOPB
PORT PWM_retry = SI_retry, DIR = out, BUS=SOPB
PORT PWM_toutSup = SI_toutSup, DIR = out, BUS=SOPB
PORT PWM_xferAck = SI_xferAck, DIR = out, BUS=SOPB

PORT PWM = "", DIR = out, VEC = [0:C_NO_CHANNELS-1]
END
```

OPB busz jelek

```
port (
  -- Global signals
  OPB_Clk : in std_logic;
  OPB_Rst : in std_logic;

  -- OPB signals
  OPB_ABus : in std_logic_vector(0 to 31);
  OPB_BE : in std_logic_vector(0 to 3);
  OPB_RNW : in std_logic;
  OPB_select : in std_logic;
  OPB_seqAddr : in std_logic;
  OPB_DBus : in std_logic_vector(0 to 31);
```

Saját jelek

```
PWM_DBus : out std_logic_vector(0 to 31);
PWM_errAck : out std_logic;
PWM_retry : out std_logic;
PWM_toutSup : out std_logic;
PWM_xferAck : out std_logic;

PWM : out std_logic_vector(0 to C_NO_CHANNELS-1);
```



PAO fájl

```
#####
# opb_core_ssp0 pao file
#####
lib proc_common_v1_00_b proc_common_pkg
lib proc_common_v1_00_b pselect
lib proc_common_v1_00_b or_muxcy
lib ipif_common_v1_00_a ipif_pkg
lib ipif_common_v1_00_a ipif_steer
lib opb_bus_attach_v1_00_a reset_mir
lib opb_bus_attach_v1_00_a opb_bus_attach
lib opb_ipif_ssp0_v1_00_a opb_ipif_ssp0

# --USER-- add all user core source files and change the following source to
# your top level core name and library

lib opb_core_ssp0_v1_00_a pwm
lib opb_core_ssp0_v1_00_a user_logic
lib opb_core_ssp0_v1_00_a opb_core_ssp0
```

Függési irány

Ezt kell módosítani



BBD fájl

- A Black Box Definition (BBD) azokat a fájlokat azonosítja, amelyeket a felhasználó specifikál perifériaegységként
- Az NGC huzalozási listát átmásolja a project/implementation könyvtárba
- Példa egyetlen fájl megadására, opciók nélkül
 - FILES
 - Blackbox.ngc
- Példa több fájl megadására, opciók nélkül
 - FILES
 - blackbox1.ngc, blackbox2.ngc, blackbox3.edn
- Példa több fájl megadására, opciókkal

C_FAMILY	C_BUS_CONFIG	FILES
virtex	1	virtex/ip1.edf
virtex	2	virtex/ip2.edf



Fájlok használata

- Két módon integrálhatjuk saját egységeinket az XPS rendszerben
 - Fekete dobozként
 - Tetszőleges egységgel szintetizáljuk
 - Az MPD és BBD fájlokkal azonosítjuk
 - Huzalozási listaként
 - Együtt szintetizáljuk a teljes processzoros rendszerrel
 - Az XST-t használjuk
 - Az MPD és PAO fájlokkal azonosítjuk





Tartalom

A Virtex-II Pro JTAG konfigurációja

© 2004 Xilinx, Inc. All Rights Reserved

JTAG TAP opciók

- Tervezési időben eldönthető, hogy az FPGA felkonfigurálása után a PowerPC processzor/processzorok JTAG TAP interfésze a az FPGA JTAG TAP interfészével közös láncba kapcsolódjon vagy maradjon független lánc
- Ezt a megfelelő processzor JTAGPPC blokkjának kihagyásával, vagy beillesztésével tehetjük meg.



JTAG TAP opciók

- Minta MHS fájl a kombinált JTAG lánc esetére

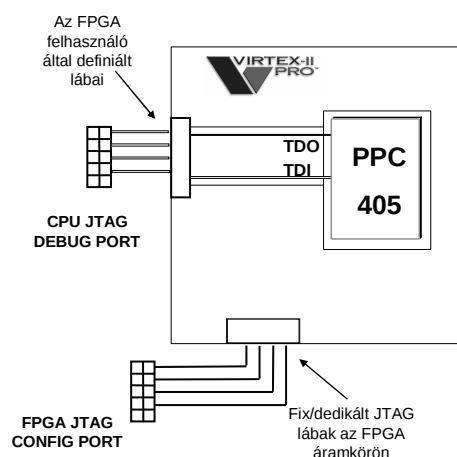
```
BEGIN jtagppc_cntlr
  PARAMETER INSTANCE = jtagppc_0
  PARAMETER HW_VER = 1.00.b
  PORT DBG405DEBUGHALT_1 = DBG405DEBUGHALT
  PORT JTGC405TRSTNEG_A11 = JTGC405TRSTNEG
  PORT JTGC405TCK_A11 = JTGC405TCK
  PORT JTGC405TDI_1 = JTGC405TDI
  PORT JTGC405TMS_A11 = JTGC405TMS
  PORT C405JGTDO_Last = C405JGTDO
  PORT C405JGTDOEN_1 = C405JGTDOEN
  PORT HALTNEG_1 = net_vcc
  PORT TRSTNEG = net_vcc
END
```

- Bekapcsolja a PowerPC™ JTAG TAP vezérlőjét az FPGA dedikált JTAG láncába a felkonfigurálás után



Virtex-II Pro Megosztott JTAG lánc

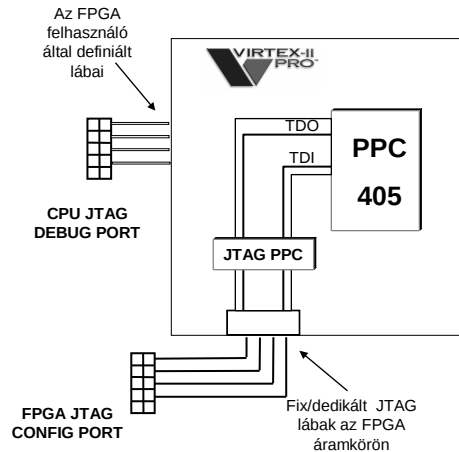
- Felhasználó által definiált JTAG lábak
 - Közvetlen, leválasztott kapcsolat a PPC405 JTAG TAP interfészhez
 - A JTAGPPC blokk nincs használva az FPGA tervében
 - A leválasztott lánc támogatja a beágyazott rendszer fejlesztési és hibakeresési eszközeit



Virtex-II Pro Egyesített JTAG lánc

- A JTAGPPC blokk használata

- Integrálja a PPC405 processzort az FPGA JTAG láncával a dedikált JTAG lábakon keresztül
- Az egyesített lánc támogatja a fejlesztő és hibakereső eszközöket
 - ChipScope™ Pro (PC4)
 - iMPACT™ (PC4)
 - GDB (PC4)
 - SingleStep™ XE (visionPROBEII®)



Virtex-II Pro Egyesített JTAG lánc

- Ha a JTAG PPC vezérlő használatával kapcsoljuk a PowerPC™ 405 CPU-t az egyesített JTAG láncba az FPGA-val, akkor minden PowerPC 405 magot összekapcsolunk egy Virtex-II Pro™ multi - CPU konfigurációba

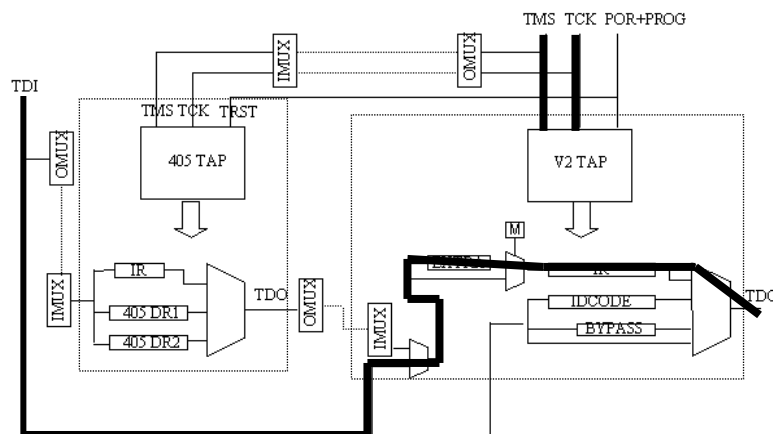


FPGA JTAG lánc összefoglalás

- PowerPC™ 405 integrált hardver CPU-k TCK, TMS, TDI, és TDO lábai konfigurálható kapcsolatok
- A Virtex-II Pro™ JTAG TAP IR utasításregisztere 6 bites, ehhez adódik hozzá a CPU-k 4 – 4 - bites utasításregisztere
 - Összesen 10 egy CPU esetén — pl. 2VP4/7
 - Összesen 14 két CPU esetén — pl. 2VP20/30/40/50/70/100
 - Összesen 22 négy CPU — pl. 2VP125
- A JTAG TAP utasításregiszter szélességet azonos értéken kell tartani az FPGA felkonfigurálása előtt és után
- Egy EXTRA regiszter kompenzálja a hiányzó CPU IR regiszter biteket a konfiguráció előtt és után, ha a CPU JTAG lábait általános I/O lábakra kötjük (amikor a JTAGPPC blokkot nem használjuk)

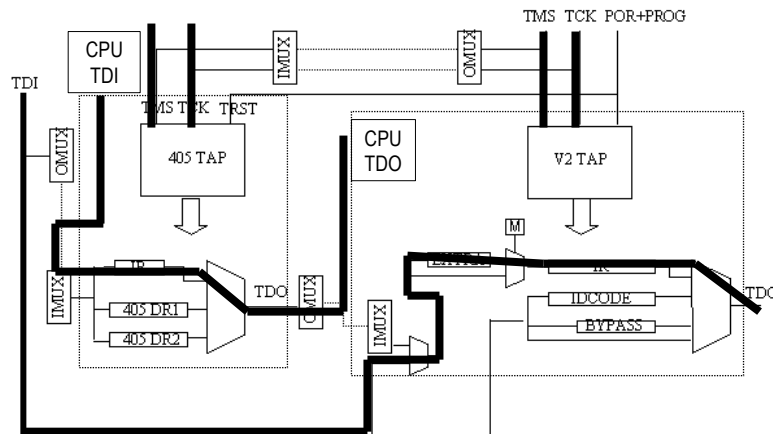


JTAG lánc kapcsolatok AZ FPGA konfigurációja előtt



JTAG lánc kapcsolatok

Konfiguráció után / Szétválasztott JTAG lánc



Hardware Design - 3 - 40

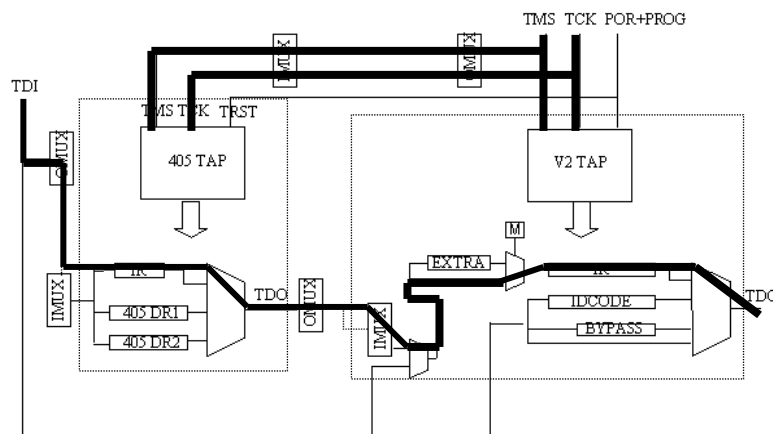
© 2004 Xilinx, Inc. All Rights Reserved

Kizárólag oktatási célra



JTAG lánc kapcsolatok

Konfiguráció után / Egyesített JTAG lánc



Hardware Design - 3 - 41

© 2004 Xilinx, Inc. All Rights Reserved

Kizárólag oktatási célra



JTAG konfiguráció

Indítás az egyesített JTAG láncról

- Az egyesített JTAG lánc lehetővé teszi a felkonfigurált FPGA CPU és külső memóriák azonos időben történő indítását
- A JTAG interfészen keresztül a rendszer konfigurációs lépései a következők lehetnek:
 - Az FPGA felkonfigurálása
 - Várakozás, amíg az FPGA kilép a konfigurációs módból
 - A CPU, CPU cache memóriák, az OCM, és a CPU által elérhető külső memóriák konfigurálása
 - A CPU PC regiszter beállítása a programkód kezdőcímére
 - Programkód végrehajtása
- A JTAG parancsok, mint SVF utasítások lehetnek a konfigurációs memóriában

