

```

=====
File :      DDR2_liesmich.pdf
Projekt :   Prj_12_DDR2
Zweck :    DDR2-SDRAM auf dem Spartan-3A Board
DDR2-RAM : MT47H32M16 (64 MByte)
Datum:     19.08.2011
Version :  7.0
Plattform : XILINX Spartan-3A
FPGA :     XC3S700A-FGG484
Sprache :  VHDL
ISE :      ISE-Design-Suite V:13.1
IP-Core :  MIG V:3.6.1
Autor :    UB
Mail :     Becker_U(at)gmx.de
=====

```

Module :

=====

```

#####
# # # # #
# DDR2-RAM : #-----# TOP_Modul_VHDL #----# Buttons_VHDL #
# 512 MBit # # # # #
# # # # #
##### # #
# # # # #
# # # #----# Clock_VHDL #
# # # #
# # # #
##### # #
# # # # #
# Eingabe : # # # # DDR2_Control_VHDL #----# DDR2_READ_VHDL #
# 4 Buttons #-----# # # # #
# 4 Schalter # # #----# # #
# # # # #
##### # # # #
# # # # #----# DDR2_Write_VHDL #
# # # # # #
# Ausgabe : #----# # # #
# 8 LEDs (Data) # # # #
# 1 LED (Status) # # # #
##### # #
# # # #----# #
##### # # DDR2_RAM_CORE.vhd #
# Clock : #-----# # # MIG 3.6.1 #
# 133 MHz # # # # (27 Dateien) #
# # # # # # + UCF-File #
##### #####

```

Zweck :  
=====

> Dieses Projekt ist ein Beispiel wie ein DDR2-SDRAM auf einem Xilinx FPGA Board eingebunden wird

Hinweise zur DDR2-RAM-Einbindung :

=====

- > das ganze Projekt basiert auf der Einbindung der vom MIG generierten Files für ein DDR2-SDRAM (MIG 3.6.1)
  
- > Hinweis zu den Einstellungen vom MIG :
  - Typ = DDR2-SDRAM
  - Frq = 133MHz
  - Write Pipe Stages = 4
  - Memory Part = MT47H32M16XX-3 (für das Spartan-3A Board)
  - Data Width = 16
  - Data-Mask = Ja
  - SystemClock = Single-Ended
  - Signale auf : Bank3 (komplett) , Bank2 (V12)
  - Bank2 = System-Clock / Bank3 = Adrees-Control+Data+System-Control
  - sonst alles auf "Default"
  
- > Hinweise zum DDR2\_RAM\_CORE :
  - es wurden nur die VHDL-Files vom Ordner "User\_Design/RTL" benutzt.
  - alle anderen vom MIG generierten Files sind für das Projekt nicht notwendig
  
- > Hinweis zum UCF-File für das DDR2-RAM :
  - ohne die Richtigen Settings im UCF-File, kommt es zu Timing problemen beim lesen oder schreiben
  - das vom MIG generierte UCF muss auf die Pinbelegung vom Spartan-3A Board angepasst werden
  - aus dem Grund habe ich das UCF-File aus dem Xilinx-Example für Spartan-3A Boards benutzt und für das Projekt angepasst das DDR2-Example bekommt man durch einen Link vom MIG -> "Xilinx Reference Boards" unter "[http://www.xilinx.com/products/boards/s3astarter/reference\\_designs.htm](http://www.xilinx.com/products/boards/s3astarter/reference_designs.htm)" der Name ist "DDR2 MIG Reference Design"

Beschreibung der Funktion :

=====

- > der Schalter-0 (SW0) dient als Reset-Schalter (High = Reset)
  
- > das "TOP\_Modul" routet alle Signale zwischen den Modulen
  
- > die "Buttons\_VHDL" dient nur zum entprellen der Tasten und zum erzeugen eines "On-Click-Events" damit die Read/Write-Funktionen nur einmal gestartet werden
  
- > das "DDR2\_Control" enthaellt eine State-Machine mit folgenden Funktionen :
  1. Initialisieren vom DDR2-RAM
  2. Beschreiben des DDR2-RAM mit 16 Datenwerten (mit je 64Bit breite)
  3. Auslesen des ersten Datenwertes
  4. Warten auf Tasteneingaben
  - 5a. Bei Taste 1 (North) -> erhoehen des Adresszeigers (ROW) um 1
  - 5b. Bei Taste 2 (South) -> verringern des Adresszeigers (ROW) um 1

- 5c. Bei Taste 3 (West) -> schreiben eines einzelnen (fixen) Datenwertes in die aktuelle Adresse
- 5d. Bei Taste 4 (East) -> lesen eines einzelnen Datenwertes aus der aktuellen Adresse

> das "DDR2\_Control" uebernimmt auch die Aufbereitung fuer die Anzeige :

- SW1 bis SW3 dient zum auswählen eines der 8 Bytewerten des 64Bit breiten Datenwortes

SW3=0 + SW2=0 + SW1=0 -> Anzeige von Datenbits (D7...D0)  
SW3=0 + SW2=0 + SW1=1 -> Anzeige von Datenbits (D15...D8)  
SW3=0 + SW2=1 + SW1=0 -> Anzeige von Datenbits (D23...D16)  
SW3=0 + SW2=1 + SW1=1 -> Anzeige von Datenbits (D31...D24)  
SW3=1 + SW2=0 + SW1=0 -> Anzeige von Datenbits (D39...D32)  
SW3=1 + SW2=0 + SW1=1 -> Anzeige von Datenbits (D47...D40)  
SW3=1 + SW2=1 + SW1=0 -> Anzeige von Datenbits (D55...D48)  
SW3=1 + SW2=1 + SW1=1 -> Anzeige von Datenbits (D63...D56)

das ausgewählte Byte wird an den 8 LEDs angezeigt

- > das "DDR2\_Read" enthaelt eine State-Machine um ein Datenwort (64Bit) aus dem RAM zu lesen
- > das "DDR2\_Write" enthaelt eine State-Machine um ein Datenwort (64Bit) ins RAM zu schreiben

Ram Inhalt nach dem Initialisieren :  
=====

> nach der Init-Funktion stehen 16 Datenwerte (je 64Bit) im RAM

ADR 0 = 0123456789ABCDEF  
ADR 1 = 123456789ABCDEF0  
ADR 2 = 23456789ABCDEF01  
ADR 3 = 3456789ABCDEF012  
ADR 4 = 456789ABCDEF0123  
ADR 5 = 56789ABCDEF01234  
ADR 6 bis ADR 15 = 639CC6398C7318E7

Ablauf :  
=====

- > nach dem Init und Schreiben von 16 Werten wird die Adr 0 ausgelesen und an den LEDs ausgegeben
- > Mit dem Button North und South kann der ADR-Counter verändert werden
- > Button EAST liest das RAM an der aktuellen Adresse aus
- > Button WEST schreibt den 64Bit-Wert "31CE629DC43B8877" in die aktuelle Adresse

Infos zum RAM :  
=====

- > das DDR2-RAM auf dem Spartan-3A Board hat eine groesse von 512MBit (64MByte)
- > das RAM ist in 4 Bloecke (BANKs) unterteilt, jeder Block ist in 8192 Reihen (ROws) und 1024 Zeilen (COLs) aufgeteilt

eine einzelne Zelle ist 16bit breit

das ergibt  $4 \times 8192 \times 1024 \times 16 \text{bit} = 512 \text{MBit}$

- > fuer die BANK\_Adresse werden 2Bit benoetigt
- fuer die ROW\_Adresse werden 13Bit benoetigt
- fuer die COL\_Adresse werden 10Bit benoetigt

> der Adresszeiger fuer eine einzelne Zelle setzt sich so zusammen :

ADR = ROW & COL & BANK

damit ist der Adresszeiger  $13+10+2 = 25 \text{bit}$  breit

Einschraenkungen in dem Projekt :

=====

- > VORSICHT !! der BURST-MODE ist fest auf "4" eingestellt, damit werden bei jedem Write Zyklus 4 Zellen (also  $4 \times 16 \text{Bit} = 64 \text{Bit}$ ) beschrieben !!
- > damit keine Daten überschrieben werden, muss der COL-Pointer immer in 4er Schritten erhöht/verringert werden (0,4,8,12 usw)

Geschwindigkeit :

=====

- > bei einer Speedmessung (per VHDL-Code) wurde die Anzahl der notwendigen Clockzyklen ermittelt
- > Read (von einem 64Bit Wert) = 22 Clockzyklen (bei 133MHz -> 165 ns => 46 MByte/sec)
- > Write (von einem 64Bit Wert) = 25 Clockzyklen (bei 133MHz -> 188 ns => 40 MByte/sec)

Hinweise :

=====

- > das Projekt ist von "privat" und nicht fuer kommerzielle Zwecke gedacht
- > das Projekt ist nicht frei von Fehlern und ich kann daher keine Garantie fuer eventuell auftretende Schaeden geben
- > das Projekt ist frei kopier- und verwendbar
- > bei gefundenen Fehlern bitte eine EMail an mich