

הכרות ותפעול עם ממשק AXI

שלב א יצירת CORE וביצוע סימולציה:

1. פתח פרויקט חדש
2. פתח את IP CATLOG
3. הגדר את זיכרון עם תצורת ממשק AXI לפי הפרמטרים הבאים:
4. בחר את הממשק לזיכרון כ- AXI

Component Name: blk_mem_gen_2

Basic	Port A Options	Port B Options	Other Options	Summary
<p>Interface Type: Native</p> <p>Memory Type: Native</p> <p>ECC Options</p>			<p><input type="checkbox"/> Generate address interface with 32 bits</p> <p><input type="checkbox"/> Common Clock</p>	

5. לטובת הסימולציה בחר את הממשק כ- AXI (שימו לב בתרגיל בפועל בשלב ב נעבוד AXI4LITE)

Component Name: blk_mem_gen_0

Basic	AXI4	Port A Options	Port B Options	Other Options	Summary
<p>AXI Type</p> <p><input checked="" type="radio"/> AXI4 <input type="radio"/> AXI4 Lite</p> <p>AXI Slave Type</p> <p><input checked="" type="radio"/> Memory Slave <input type="radio"/> Peripheral Slave</p> <p>ID Width Configuration</p>					

1. הגדר את הזיכרון כ-simple dual ram

Component Name		blk_mem_gen_2		
Basic	Port A Options	Port B Options	Other Options	Summary
Interface Type	Native	<input type="checkbox"/> Generate address interface with 32 bits		
Memory Type	Simple Dual Port RAM	<input type="checkbox"/> Common Clock		

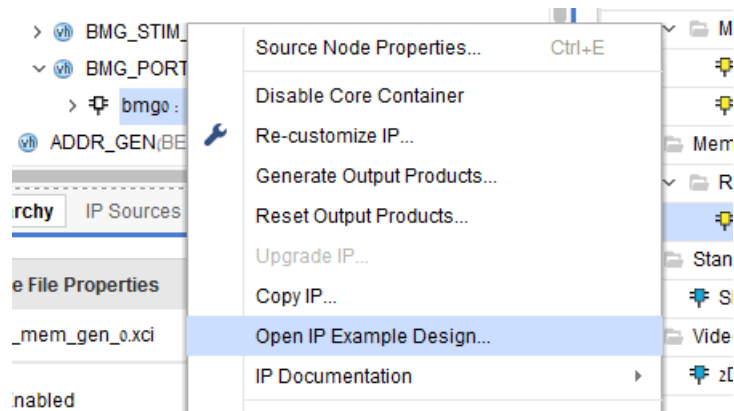
2. הגדר את PORTA כרוחב 32 ביט ל1024 תאים

Component Name		blk_mem_gen_0			
Basic	AXI4	Port A Options	Port B Options	Other Options	Summary
Memory Size					
Port A Width	32	Range: 32 to 256 (bits)			
Port A Depth	1024	Range: 1024 to 1048576			
The Width and Depth values are used for Write Operations in Port A					
Operating Mode	Read First	Enable Port Type	Use ENA Pin		

3. הגדר את PORTB כרוחב 32 ביט ל1024 תאים
































Component Name		blk_mem_gen_0			
Basic	AXI4	Port A Options	Port B Options	Other Options	Summary
Memory Size					
Port B Width	32				
Port B Depth	1024				
The Width and Depth values are used for Read Operation in Port B					
Operating Mode	Read First	Enable Port Type	Use ENB Pin		

4. עמוד על CORE שיצרת והרץ את OPEN IP EXMAPLE (הגדל זמן ריצה בצורה ניכרת ל- 1MS)



בחר צורות גלים שמתארות את ממשק ה AXI לפי הרשימה לעיל והרץ שוב

Name	Design Unit	Block Type
✓ blk_mem_gen_g...	blk_mem_g...	VHDL Entity
> blk_mem_ge...	blk_mem_g...	VHDL Entity
glbl	glbl	Verilog M...

Objects	
	
Name	
 CLK_IN	
 RESET_IN	
>  STATUS[6:0]	
 CLKA	
 RSTA	
 S_ACLK	
 S_ARESETN	
>  S_AXI_AWID[3:0]	
>  S_AXI_AWADDR[31:0]	
>  S_AXI_AWLEN[7:0]	
>  S_AXI_AWSIZE[2:0]	
>  S_AXI_AWBURST[1:0]	
 S_AXI_AWVALID	
 S_AXI_AWREADY	
>  S_AXI_WDATA[31:0]	
>  S_AXI_WSTRB[3:0]	
 S_AXI_WLAST	
 S_AXI_WVALID	
 S_AXI_WREADY	
>  S_AXI_BID[3:0]	
>  S_AXI_ARID[3:0]	
>  S_AXI_RID[3:0]	
>  S_AXI_BRESP[1:0]	
 S_AXI_BVALID	
 S_AXI_BREADY	
>  S_AXI_ARLEN[7:0]	
>  S_AXI_ARSIZE[2:0]	
>  S_AXI_ARBURST[1:0]	
 S_AXI_RLAST	

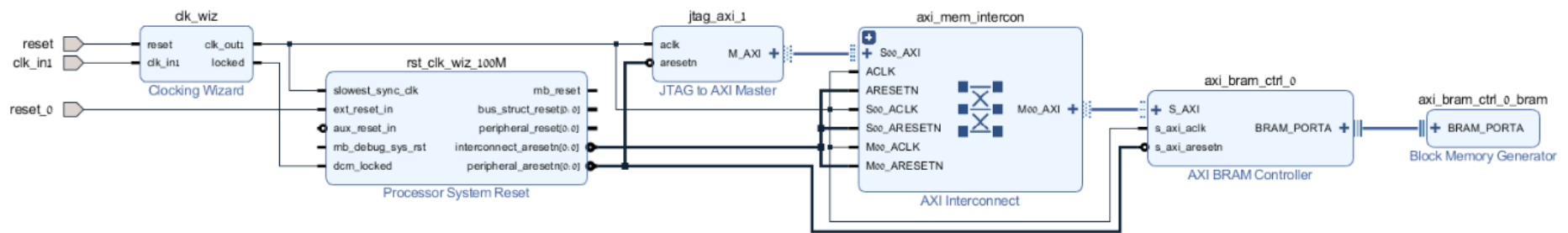
The timing diagram displays the behavior of the blk_mem_gen_0 block over a 30 microsecond period. A yellow vertical line indicates a specific time point at 2.150000 us. The signals shown include:

- RESET**: Active low signal, transitions from high to low at the start.
- S_AXI_ACLK**: System clock signal, shown as a green periodic waveform.
- S_ARESETN**: Active low reset signal, transitions from low to high.
- S_AXI_AWADDR[31:0]**: Address on the write channel, showing a sequence of addresses (0, 2, 4, 5, 6, 7, 8, 9, 10, 11, 12).
- S_AXI_AWVALID**: Valid signal for the write address, active during address periods.
- S_AXI_AWREADY**: Ready signal for the write address, active during address periods.
- S_AXI_WDATA[31:0]**: 32-bit data on the write channel, showing hexadecimal values like 00000000, 00000001, 00000002, etc.
- S_AXI_AWLEN[7:0]**: Length of the write address, mostly 0.
- S_AXI_WLAST**: Last signal for the write address, active at the end of each address period.
- S_AXI_WVALID**: Valid signal for the write data, active during data periods.
- S_AXI_WREADY**: Ready signal for the write data, active during data periods.
- S_AXI_BRESP[1:0]**: Response signals for the write channel, mostly 0.
- S_AXI_BVALID**: Valid signal for the write data, active during data periods.
- S_AXI_BREADY**: Ready signal for the write data, active during data periods.
- S_AXI_AWSIZE[2:0]**: Size of the write address, mostly 2.
- S_AXI_AWBURST[1:0]**: Burst type for the write address, mostly 1.
- S_AXI_WSTRB[3:0]**: Write strobe signal, active during data periods.
- S_AXI_BID[3:0]**: ID signal for the write channel, mostly 0.
- S_AXI_ARID[3:0]**: ID signal for the read channel, mostly 0.
- S_AXI_RID[3:0]**: ID signal for the read channel, mostly 0.
- S_AXI_ARLEN[7:0]**: Length of the read address, mostly 0.
- S_AXI_ARSIZE[2:0]**: Size of the read address, mostly 2.
- S_AXI_ARBURST[1:0]**: Burst type for the read address, mostly 1.
- S_AXI_RLAST**: Last signal for the read address, active at the end of each address period.
- S_AXI_ARADDR[31:0]**: Address on the read channel, showing a sequence of addresses (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10).
- S_AXI_ARVALID**: Valid signal for the read address, active during address periods.
- S_AXI_ARREADY**: Ready signal for the read address, active during address periods.
- S_AXI_RDATA[31:0]**: 32-bit data on the read channel, showing hexadecimal values like 0, 336, 1346, 25244, 5958, 3288679298, 2568914959.
- S_AXI_RRSP[1:0]**: Response signals for the read channel, mostly 0.
- S_AXI_RVALID**: Valid signal for the read data, active during data periods.
- S_AXI_RREADY**: Ready signal for the read data, active during data periods.

בס"ד

שלב ב

1. בנה בעזרת ה-IP INETGARTOR את התכנון הבא



2. נא וודא שאין טעויות תכן בעזרת VALIDATE

3. נא הוסף את קובץ XDC בהתאם (bh)

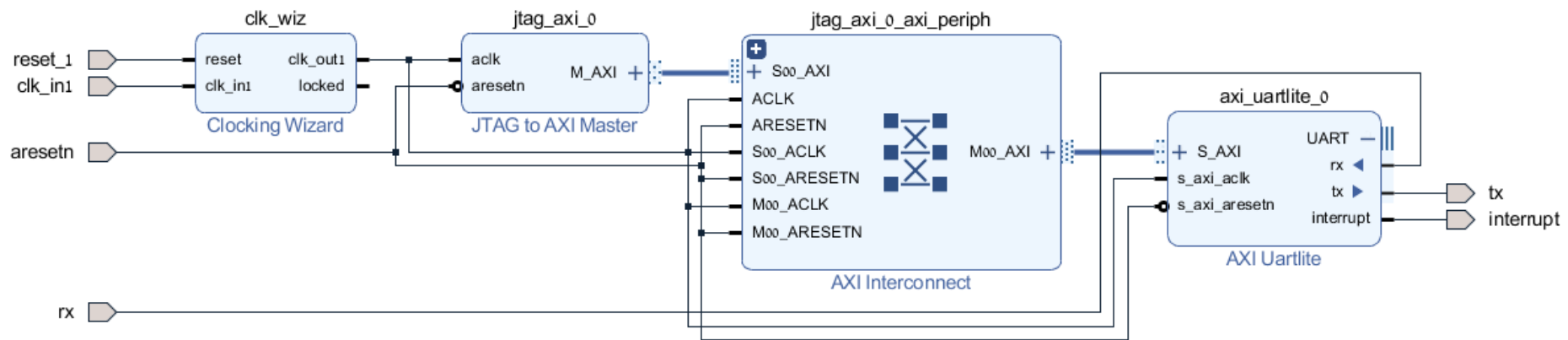
4. ב chip scope נא להוסיף את הסיגנלים של ממשק AXI-4 LITE

5. הרץ את SCRIPT הבא (bh_demo)

בס"ד

שלב ג

1. בנה בעזרת ה-IP INETGARTOR את התכנון הבא



2. נא וודא שאין טעויות תכן בעזרת VALIDATE

3. נא הוסף את קובץ XDC בהתאם (bh_1lb)

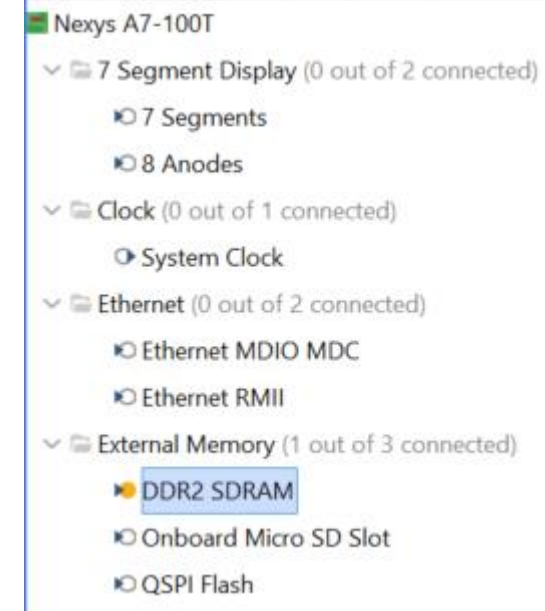
4. הרץ את SCRIPT הבא: (bh_demo_uart_transsmmit)

5. נא בדוק בטרנימל שאכן מתקבל השידור ששלחת (ABCDE)

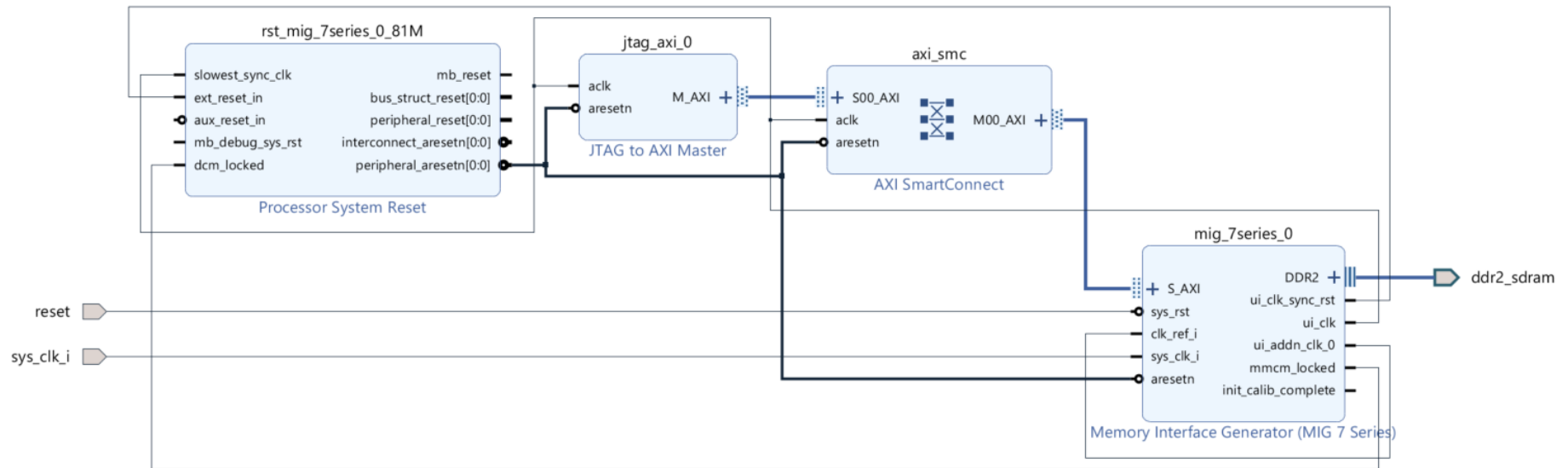
בס"ד

שלב ד

1. בנה בעזרת ה-IP INETGARTOR את התכנון הבא
2. נא הוסף את ממשק DDR2 בעזרת ה-TAB הבא



3.



1. נא וודא שאין טעויות תכן בעזרת VALIDATE
2. נא בדוק למול הקובץ XDC בהתאם (bh_dds_demo_warpper)
3. הרץ את SCRIPT הבא: (bh_demo_DDR2)
4. נא בדוק

חומרים לשימוש פנימי בלבד אין להעביר ללא רשות מניר בלולו