

ΑΡΙΣΤΟΤΕΛΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ & ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΗΥ1901– Ειδικές Αρχιτεκτονικές Υπολογιστών

Εργαστήριο 2
Οργάνωση Υλικού με το Vivado

Ν. ΤΑΜΠΟΥΡΑΤΖΗΣ - Π. ΜΟΥΣΟΥΛΙΩΤΗΣ

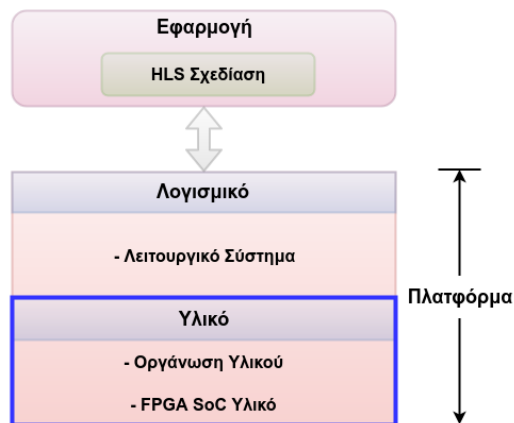
Διδάσκων: Ιωάννης Παπαευσταθίου

Version 0.1

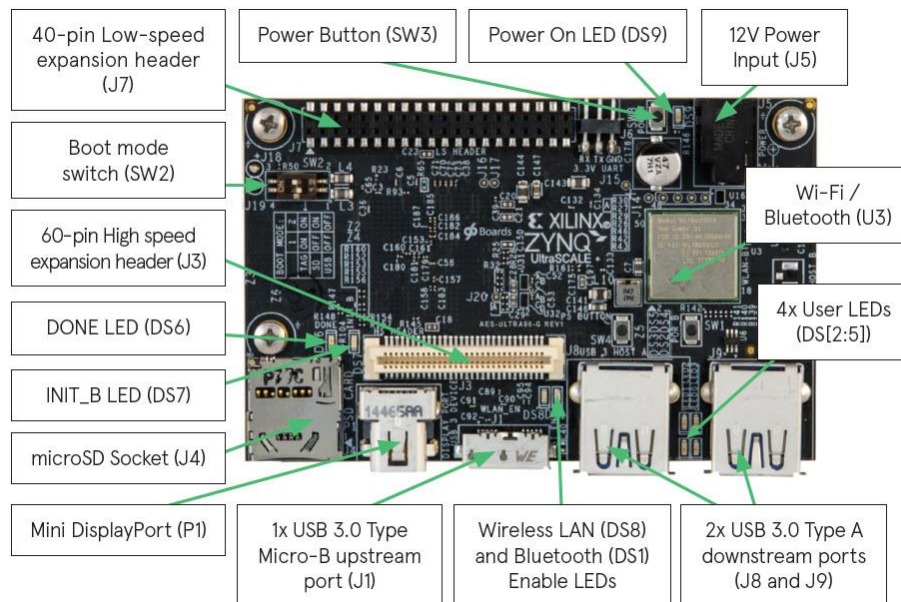
Νοέμβριος 2019

1. Εισαγωγή

Στο εργαστήριο 1 ασχοληθήκαμε με τη σχεδίαση του επιταχυντή μας με τη χρήση του εργαλείου Vivado HLS. Στο παρόν εργαστήριο θα οργανώσουμε το υλικό μέρος της πλατφόρμας που θα χρησιμοποιήσουμε για να διασυνδέσουμε τον επιταχυντή μας. Η πλατφόρμα αποτελείται από υλικό και λογισμικό συστήματος όπως φαίνεται στο παρακάτω σχήμα. Στο παρόν εργαστήριο θα ασχοληθούμε με το τμήμα που περικλείεται στο μπλε πλαίσιο, το “Υλικό” της πλατφόρμας.



Το υλικό της πλατφόρμας αποτελείται από το FPGA SoC που θα χρησιμοποιήσουμε, το **xczu3eg-sbva484-1-e**, το οποίο βρίσκεται πάνω στην πλακέτα Ultra96v1 (version 1.2). Στο πιο κάτω σχήμα φαίνεται η πλακέτα Ultra96v1 που περιλαμβάνει το FPGA SoC **xczu3eg-sbva484-1-e** (το FPGA SoC δεν φαίνεται στο σχήμα γιατί βρίσκεται στην άλλη πλευρά της πλακέτας).



Η οργάνωση του υλικού θα γίνει με τη χρήση του εργαλείου Vivado της Xilinx και αφορά:

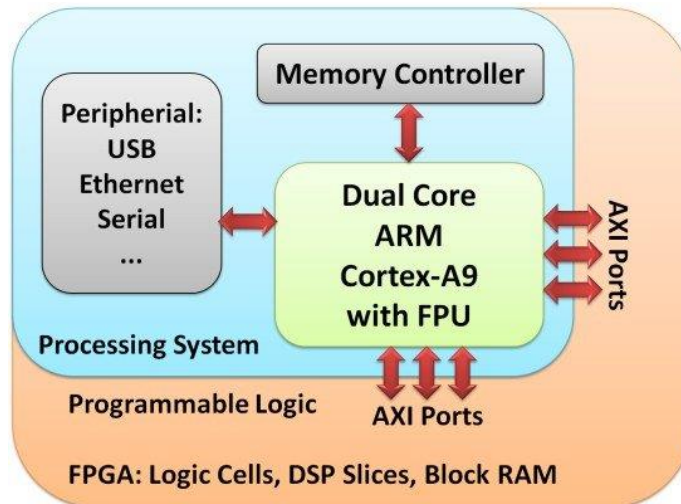
- τον καθορισμό του FPGA SoC και της πλακέτας που θα χρησιμοποιήσουμε
- την επιλογή, την παραμετροποίηση και τη διασύνδεση τμημάτων του FPGA SoC και της πλακέτας έτσι ώστε να εξυπηρετεί τη σχεδίασή μας

Σε αυτό το εργαστήριο θα εξοικειωθούμε με τη σουίτα σχεδίασης Vivado¹ της Xilinx (Vivado Design Suite) και πιο συγκεκριμένα με τη δυνατότητα σχεδίασης με IP² blocks της σουίτας (IP Integrator).

Η οργάνωση του υλικού (από εδώ και πέρα θα την αποκαλούμε και ως η “σχεδίασή” μας) θα αποτελεί μία Zynq® UltraScale+ MPSoC σχεδίαση που θα περιλαμβάνει έναν ARM ως σύστημα επεξεργασίας (processing system - PS) καθώς και IP blocks στο τμήμα προγραμματιζόμενης λογικής (programmable logic - PL). Στο πιο κάτω σχήμα φαίνεται η απλοποιημένη αναπαράσταση της αρχιτεκτονικής Zynq των FPGA SoCs της Xilinx.

¹ https://www.xilinx.com/support/documentation/sw_manuals/xilinx2018_2/ug910-vivado-getting-started.pdf

² <https://whatis.techtarget.com/definition/IP-core-intellectual-property-core>



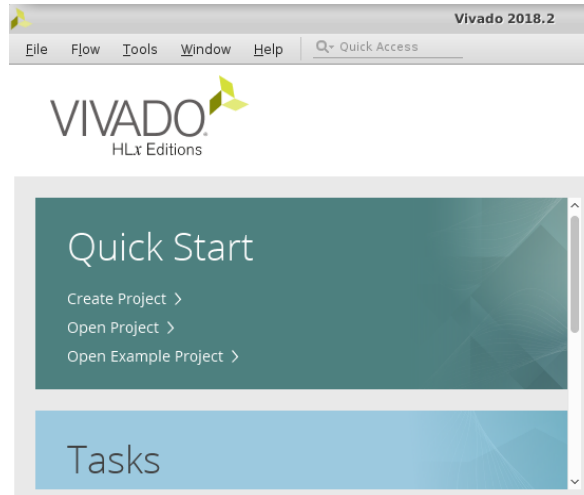
Κάνοντας χρήση της δυνατότητας IP Integrator του Vivado, θα εισάγουμε στη σχεδίασή μας τα εξής IPs:

- IP για διαχείριση σήματος ρολογιού (Clocking Wizard IP) το οποίο θα χρησιμοποιηθεί για την παραγωγή σημάτων ρολογιού με διαφορετικές συχνότητες για χρήση από τον επιταχυντή μας
- IP για σύγχρονη επανεκκίνηση πηγών σημάτων ρολογιού (Processor System Reset IP)
- IP για συγχώνευση όλων των πηγών σημάτων διακοπής από μπλοκ στο PL προς το PS (Concat IP)

Επίσης, μέσω του Vivado θα γίνει επιλογή των διαθέσιμων διεπαφών επικοινωνίας μεταξύ του PS και του επιταχυντή στο PL. Στο τέλος, η IP σχεδίαση μαζί με την πληροφορία που αφορά τις διεπαφές επικοινωνίας θα αποθηκευτούν σε ένα αρχείο (το αρχείο υποστήριξης συσκευής - Device Support Archive - DSA) το οποίο θα χρησιμοποιηθεί για να τον ορισμό της πλατφόρμας η οποία με τη σειρά της θα χρησιμοποιηθεί από το εργαλείο SDSoC (Εργαστήριο 3).

2. Δημιουργία νέου project στο Vivado

Αφού ανοίξετε το Vivado IDE, επιλέξτε στο Quick Start το **Create Project** όπως φαίνεται στην πιο κάτω εικόνα.



Πατήστε **Next** στο παράθυρο που θα ανοίξει για να αρχίσετε τη δημιουργία του καινούριου project. Στο παράθυρο που ακολουθεί (New Project), δώστε ως όνομα του project το **ultra96v1** και επιλέξτε την τοποθεσία που επιθυμείτε να αποθηκευτεί.

Project Name
Enter a name for your project and specify a directory where the project data files will be stored.

Project name:

Project location: ...

☒ Create project subdirectory

Project will be created at: .../projects/vivado_2018.2/ultra96v1

Στο παράθυρο Project Type επιλέξτε **RTL Project**, επιλέξτε το **Do not specify sources at this time** και πατήστε στο **Next** για να δημιουργήσετε ένα project στο οποίο η σχεδίαση θα γίνει με IP blocks με τη χρήση του IP Integrator χωρίς την εισαγωγή πηγαίων αρχείων (source files).

Project Type

Specify the type of project to create.



RTL Project

You will be able to add sources, create block designs in IP Integrator, generate IP, run RTL analysis, synthesis, implementation, design planning and analysis.



Do not specify sources at this time



Post-synthesis Project: You will be able to add sources, view device resources, run design analysis, planning and implementation.



< Back

Next >

Finish

Cancel

Στο παράθυρο Default Part επιλέξτε το **Boards**, στο πεδίο Search πληκτρολογήστε το **ultra96v1** και κάντε κλικ πάνω στην πλακέτα που βγάζει από κάτω ως αποτέλεσμα - την **Ultra96v1 Evaluation Platform**.

Default Part

Choose a default Xilinx part or board for your project. This can be changed later.



Parts | **Boards**

[Reset All Filters](#)

Vendor: All

Name: All

Board Rev: Latest

Search: ultra96v1 (1 match)

Display Name	Preview	Vendor	File Ver...	Part	I/O Pir
Ultra96v1 Evaluation Platform		em.avnet.com	1.2	xczu3eg-sbva484-1-e	484



< Back

Next >

Finish

Cancel

Το τελευταίο παράθυρο που λέγεται New Project Summary θα πρέπει να περιλαμβάνει τις ίδιες πληροφορίες με την πιο κάτω εικόνα.



New Project Summary

A new RTL project named 'ultra96v1' will be created.

The default part and product family for the new project:
Default Board: Ultra96v1 Evaluation Platform
Default Part: xczu3eg-sbva484-1-e
Product: Zynq UltraScale+
Family: Zynq UltraScale+ MPSoCs
Package: sbva484
Speed Grade: -1

To create the project, click Finish



< Back

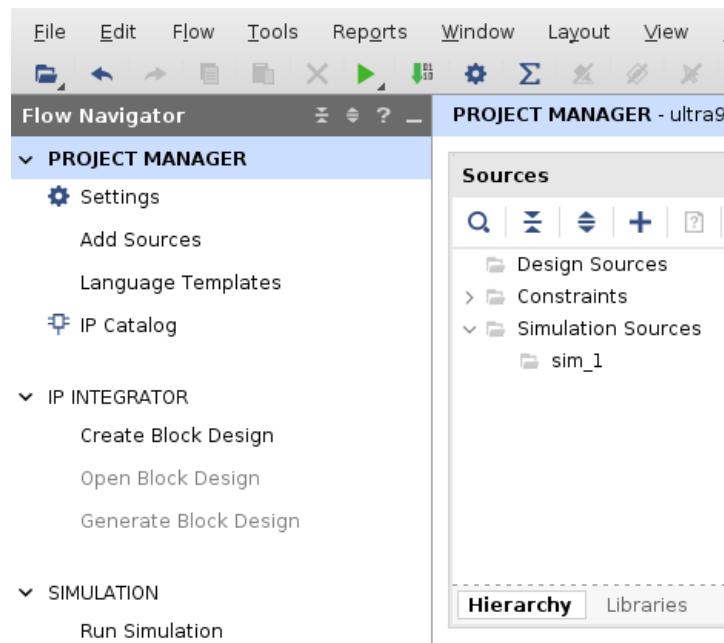
Next >

Finish

Cancel

3. Δημιουργία σχεδίασης με χρήση IP blocks

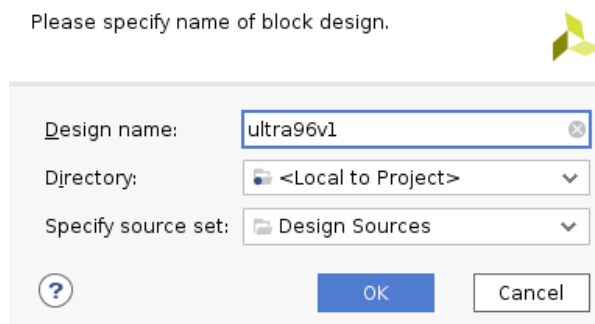
Αφού έχουμε ολοκληρώσει τη διαδικασία δημιουργίας ενός καινούριου project, το Vivado IDE μας παρουσιάζει ένα παράθυρο που αποτελείται από πολλά υπο-παράθυρα, τα οποία περιλαμβάνουν πληροφορίες σχετικά με το project μας. Ένα από αυτά τα υπο-παράθυρα, το **Flow Navigator**, παίζει καθοριστικό ρόλο στη σχεδίαση αφού περιγράφει τα διάφορα βήματα που έχει τη δυνατότητα να ακολουθήσει ένας σχεδιαστής κατά τη ροή της σχεδίασης. Επειδή η σχεδίασή μας θα γίνει με τη χρήση IP blocks, επιλέγουμε στο **Flow Navigator** το **Create Block Design** που μπορούμε να το βρούμε αφού αναπτύξουμε (expand) το εργαλείο **IP INTEGRATOR**.



Στο παράθυρο που θα ανοίξει εισάγουμε το όνομα της σχεδίασής μας.

ΠΡΟΣΟΧΗ: Το όνομα αυτό πρέπει να είναι ίδιο με το όνομα της τελικής πλατφόρμας που θα χρησιμοποιηθεί από το SDSoC.

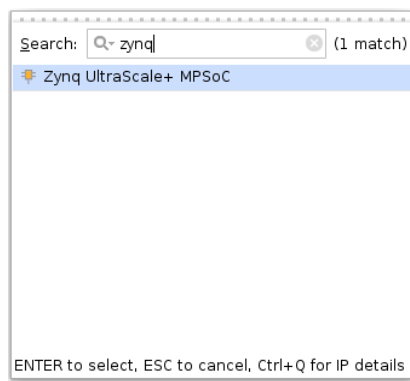
Εισάγουμε το **ultra96v1** ως όνομα και επιλέγουμε το **OK**.



Παρατηρούμε ότι το παράθυρο του Vivado IDE αλλάζει μορφή - το Flow Navigator παράθυρο παραμένει στην ίδια μορφή ενώ το PROJECT MANAGER υπό-παράθυρο δίνει τη θέση του στο BLOCK DESIGN υπό-παράθυρο. Τώρα είμαστε έτοιμοι για να αρχίσουμε τη σχεδιάσή μας. Ο πιο κάτω πίνακας δίνει μια συνοπτική περιγραφή των IP blocks που θα περιλαμβάνει η σχεδιάσή μας.

IP Block	Usage Summary
Zynq UltraScale+ MPSoC Processor System (PS)	<ul style="list-style-type: none"> • Dual-core Arm processor with cache hierarchy • Integrated I/O peripherals • DDR memory controller with external memory interface • PS to Programmable Logic (PL) interconnects • PL to PS interconnects
Processor System Reset Block (PL)	<ul style="list-style-type: none"> • Reset sequencing and synchronization block for PL logic
Clocking Wizard (PL)	<ul style="list-style-type: none"> • Multiple output clock generator to drive PL logic
Concat Block (PL)	<ul style="list-style-type: none"> • PL interrupt structure that feeds Zynq UltraScale+ MPSoC PS interrupt request input

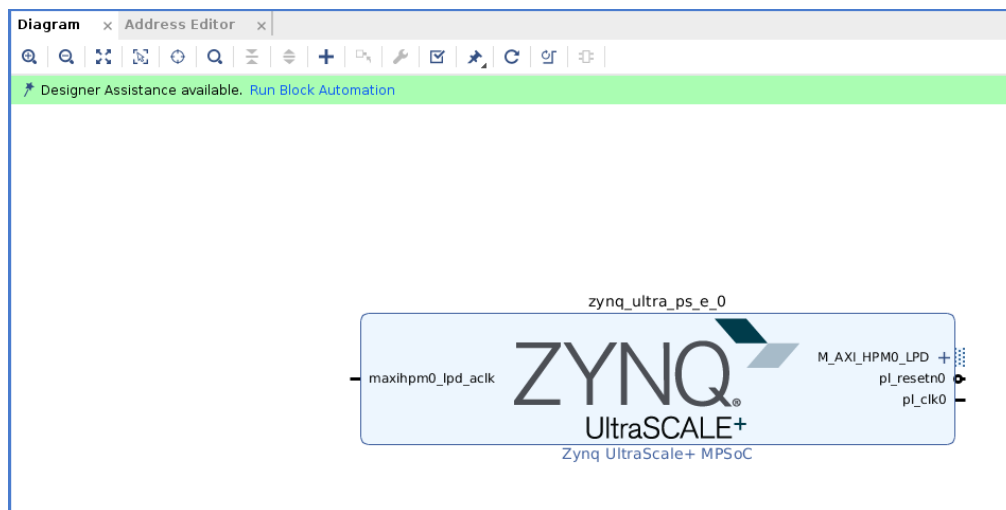
Στο τμήμα **Diagram** του BLOCK DESIGN υπό-παράθυρου, πατάμε πάνω στο σύμβολο + για να προσθέσουμε IP blocks ή κάνουμε δεξί κλικ στο **Diagram** τμήμα και επιλέγουμε **Add IP**. Στο παράθυρο που ανοίγει, εισάγουμε **zynq** στο πεδίο αναζήτησης **Search** και πατάμε **Enter** ή **διπλό κλικ** πάνω στο **Zynq UltraScale+ MPSoC** για να το εισάγουμε στο διάγραμμά μας. Με αυτόν τον τρόπο εισάγουμε το PS στη σχεδιάσή μας.



ΠΑΡΑΤΗΡΗΣΤΕ: Κάτω από το τμήμα Diagram στο υπό-παράθυρο BLOCK DESIGN υπάρχει ένα τμήμα μηνυμάτων που περιλαμβάνει ένα tab που ονομάζεται Tcl Console. Στο Tcl Console αναγράφονται όλες οι ενέργειές μας στο Vivado IDE σε μορφή εντολών. Οπότε αντί των ενεργειών μας, θα μπορούσαμε να δώσουμε ως είσοδο στο Vivado ένα script Tcl για να δημιουργήσουμε το project και τη σχεδίασή μας. Επειδή η σχεδίαση με χρήση Tcl script είναι

πιο προχωρημένη (advanced), δεν θα την ακολουθήσουμε στα πλαίσια αυτού του εργαστηρίου.

Στο τμήμα Diagram έχει εμφανιστεί ένα μήνυμα του **Designer Assistance** το οποίο λέει **Run Block Automation**. Αυτού του είδους τα μηνύματα εμφανίζονται κατά τη διάρκεια της σχεδίασης για να γίνουν αυτόματα κάποιες ενέργειες και να απλοποιηθεί η διαδικασία. Κάνοντας κλικ στο **Run Block Automation** ανοίγει ένα παράθυρο το οποίο μας ενημερώνει ότι μπορεί να διαμορφώσει το PS ώστε να ανταποκρίνεται στις διασυνδέσεις του με τα περιφερειακά της υπόλοιπης πλακέτας Ultra96v1 (**Apply Board Preset**) όπως αυτή παρουσιάστηκε σε σχήμα της παραγράφου 1 του παρόντος εργαστηρίου. Επιλέγουμε **OK** για να συνεχίσουμε τη σχεδίαση.



Automatically make connections in your design by checking the boxes of the blocks to connect. Select a block on the left to display its configuration options on the right.



Description

This option sets the board preset on the Processing System. All current properties will be overwritten by the board preset. This action cannot be undone.

NOTE: Apply Board Preset will discard existing IP configuration - please uncheck this box, if you wish to retain previous configuration.

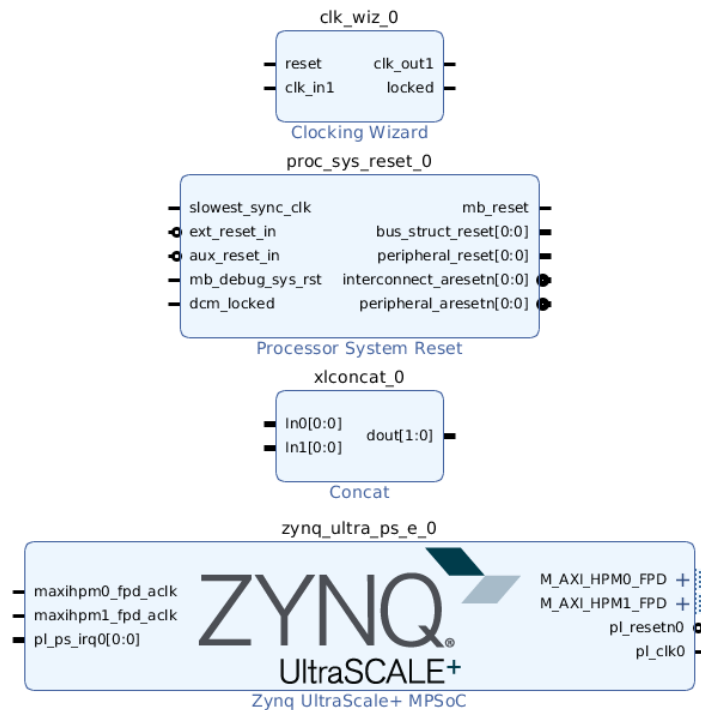
Instance: /zynq_ultra_ps_e_0

Options

Apply Board Preset: ☒

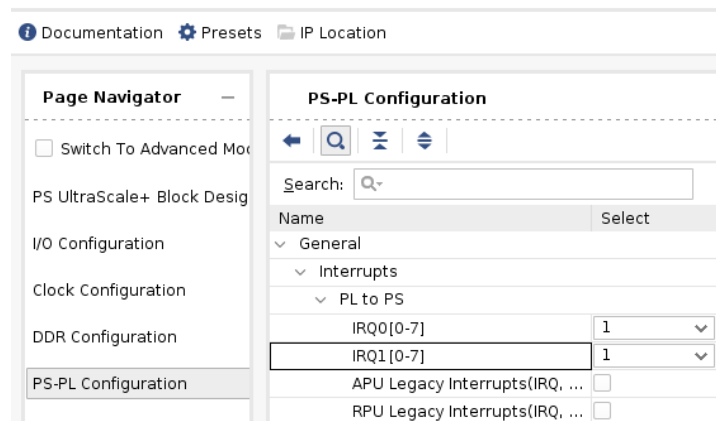
OK Cancel

Επαναλαμβάνουμε τη διαδικασία προσθήκης IP, για να προσθέσουμε τα **Processor System Reset, Clocking Wizard** και **Concat** IPs όπως φαίνεται στην παρακάτω εικόνα. Αυτά τα IPs κάνουν χρήση των πόρων του PL (FPGA) για να υλοποιηθούν.



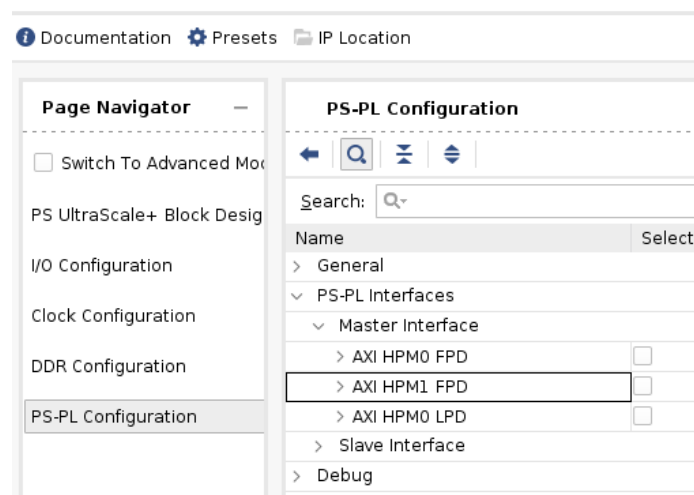
Μετά την εισαγωγή των IP blocks, θα πρέπει να ρυθμίσουμε τις παραμέτρους τους ώστε να ανταποκρίνονται στις απαιτήσεις της σχεδιάσής μας. Πρώτα θα ρυθμίσουμε το **Zynq UltraScale+ MPSoC** IP block πατώντας διπλό κλικ πάνω του. Για να ενεργοποιήσουμε τις διακοπές από το PL προς το PS επιλέγουμε το **PS-PL Configuration** που βρίσκεται στο **Page Navigator** τμήμα του παραθύρου **Re-customize IP** και αναπτύσσουμε στα δεξιά του **Page Navigator** το **General** -> **Interrupts** -> **PL to PS** και εισάγουμε τον αριθμό **1** στα **IRQ0[0-7]** και **IRQ1[0-7]**. Με αυτή μας την ενέργεια επιτρέπουμε την διαχείριση 8 ξεχωριστών σημάτων διακοπής από την κάθε interrupt είσοδο του PS. Με αυτόν τον τρόπο ενεργοποιούμε τη PS-PL διεπαφή διακοπών η οποία θα συνδεθεί με το Concat IP που βρίσκεται στο PL. Η είσοδος του Concat IP θα παραμείνει ελεύθερη για να χρησιμοποιηθεί από το SDSoC έτσι ώστε να διασυνδέσει σήματα διακοπών από τον επιταχυντή που βρίσκεται στο PL προς το PS.

Zynq UltraScale+ MPSoC (3.2)



Επιλέγουμε ξανά το **PS-PL Configuration** που βρίσκεται στο **Page Navigator** τμήμα του παραθύρου **Re-customize IP** και αναπτύσσουμε στα δεξιά του **Page Navigator** το **PS-PL Interfaces** -> **Master Interface** και αποεπιλέγουμε τα **AXI HPM0 FPD** and **AXI HPM1 FPD**. Αυτό το κάνουμε για να αποδεσμεύσουμε αυτές τις διεπαφές από το PS για να μπορούν να χρησιμοποιηθούν από τον επιταχυντή μέσω του εργαλείου SDSoc. Πατάμε το **OK** που βρίσκεται κάτω δεξιά για να κλείσουμε το παράθυρο **Re-customize IP**.

Zynq UltraScale+ MPSoC (3.2)



Κάνουμε διπλό-κλικ πάνω στο **Clocking Wizard IP** για να το παραμετροποιήσουμε. Στην καρτέλα **Clocking Options** του παραθύρου **Re-customize IP** παρατηρήστε ότι το ρολόι εισόδου (**clk_in1**) στο **Clocking Wizard IP** έχει τιμή 100MHz - αυτό το σήμα παρέχεται από το PS (είναι το σήμα **p1_clk0** του Zynq UltraScale+ MPSoC IP). Το **Clocking Wizard IP** μπορεί να χρησιμοποιηθεί για την παραγωγή σημάτων ρολογιού με συχνότητα είτε μεγαλύτερη είτε μικρότερη από τη συχνότητα του σήματος ρολογιού εισόδου (**clk_in1**). Βεβαιωθείτε ότι στην καρτέλα **Output Clocks** το σήμα **clk_out1** έχει τιμή 100MHz. Επίσης βεβαιωθείτε ότι το **Reset**

Type είναι ρυθμισμένο στην επιλογή **Active Low**. Πατάμε το **OK** που βρίσκεται κάτω δεξιά για να κλείσουμε το παράθυρο **Re-customize IP**.

Component Name clk_wiz_0

Clocking Options	Output Clocks	MMCM Settings	Summary				
<input checked="" type="checkbox"/> clk_out1	clk_out1	100.000	100.000	0.000	0.000	50.000	
<input type="checkbox"/> clk_out2	clk_out2	100.000	N/A	0.000	N/A	50.000	
<input type="checkbox"/> clk_out3	clk_out3	100.000	N/A	0.000	N/A	50.000	
<input type="checkbox"/> clk_out4	clk_out4	100.000	N/A	0.000	N/A	50.000	
<input type="checkbox"/> clk_out5	clk_out5	100.000	N/A	0.000	N/A	50.000	
<input type="checkbox"/> clk_out6	clk_out6	100.000	N/A	0.000	N/A	50.000	
<input type="checkbox"/> clk_out7	clk_out7	100.000	N/A	0.000	N/A	50.000	

☐ USE CLOCK SEQUENCING

Output Clock	Sequence Number
clk_out1	1
clk_out2	1
clk_out3	1
clk_out4	1
clk_out5	1
clk_out6	1
clk_out7	1

Enable Optional Inputs / Outputs for MMCM/PLL☒ reset☐ power_down☐ input_clk_stopped☒ locked☐ clkfbstopped

Reset Type☐ Active High☒ Active Low

Phase Shift Mode☐ WAVEFORM☒ LATENCY

Κάνουμε διπλό-κλικ πάνω στο **Concat IP** για να το παραμετροποιήσουμε. Θέτουμε την τιμή **1** στο πεδίο **Number of Ports** και πατάμε το **OK** που βρίσκεται κάτω δεξιά για να κλείσουμε το παράθυρο **Re-customize IP**. Το εργαλείο SDSoC θα διοχετεύσει διακοπές από το PL προς το PS μέσω του **Concat IP**.

Στη συνέχεια επιλέγουμε το **Run Connection Automation** του **Designer Assistance** (το μήνυμα εμφανίζεται κάτω από το τμήμα **Diagram** του παραθύρου **BLOCK DESIGN**). Η ενέργεια αυτή μας βοηθάει στο να διασυνδέσουμε τα **Clocking Wizard** και **Processor System Reset IP blocks** με το **Zynq UltraScale+ MPSoC IP block**. Το σήμα **clk_in1** του **Clocking Wizard IP** θα πρέπει να συνδεθεί με το **pl_clk0** του PS και το **slowest_sync_clk setting** του **Processor System Reset IP** θα πρέπει να συνδεθεί με το **clk_out1** του **Clocking Wizard IP**.

4. Προσθήκη ρολογιών, σημάτων Reset και διακοπών

Η μέχρι τώρα σχεδίασή μας περιλαμβάνει για τον accelerator 1 ρολόι, 1 σήμα reset και 8 σήματα διακοπής. Σε αυτή τη παράγραφο θα προσθέσουμε σήματα ρολογιού, reset και διακοπών για να μπορεί στο τέλος η πλατφόρμα μας να υποστηρίξει εφαρμογές που έχουν περισσότερες απαιτήσεις χωρίς να χρειάζεται να την ξανα-δημιουργήσουμε για να προσθέσουμε αυτές τις δυνατότητες.

Για το σκοπό αυτό παραμετροποιήστε το **Clocking Wizard** IP έτσι ώστε να παράγει ακόμα 6 ρολόγια με τις εξής συχνότητες: 75, 150, 200, 300, 400, 600 MHz.

Εισάγετε ακόμα 6 **Processor System Reset** IPs (hint: μπορείτε να κάνετε copy-paste το υπάρχον).

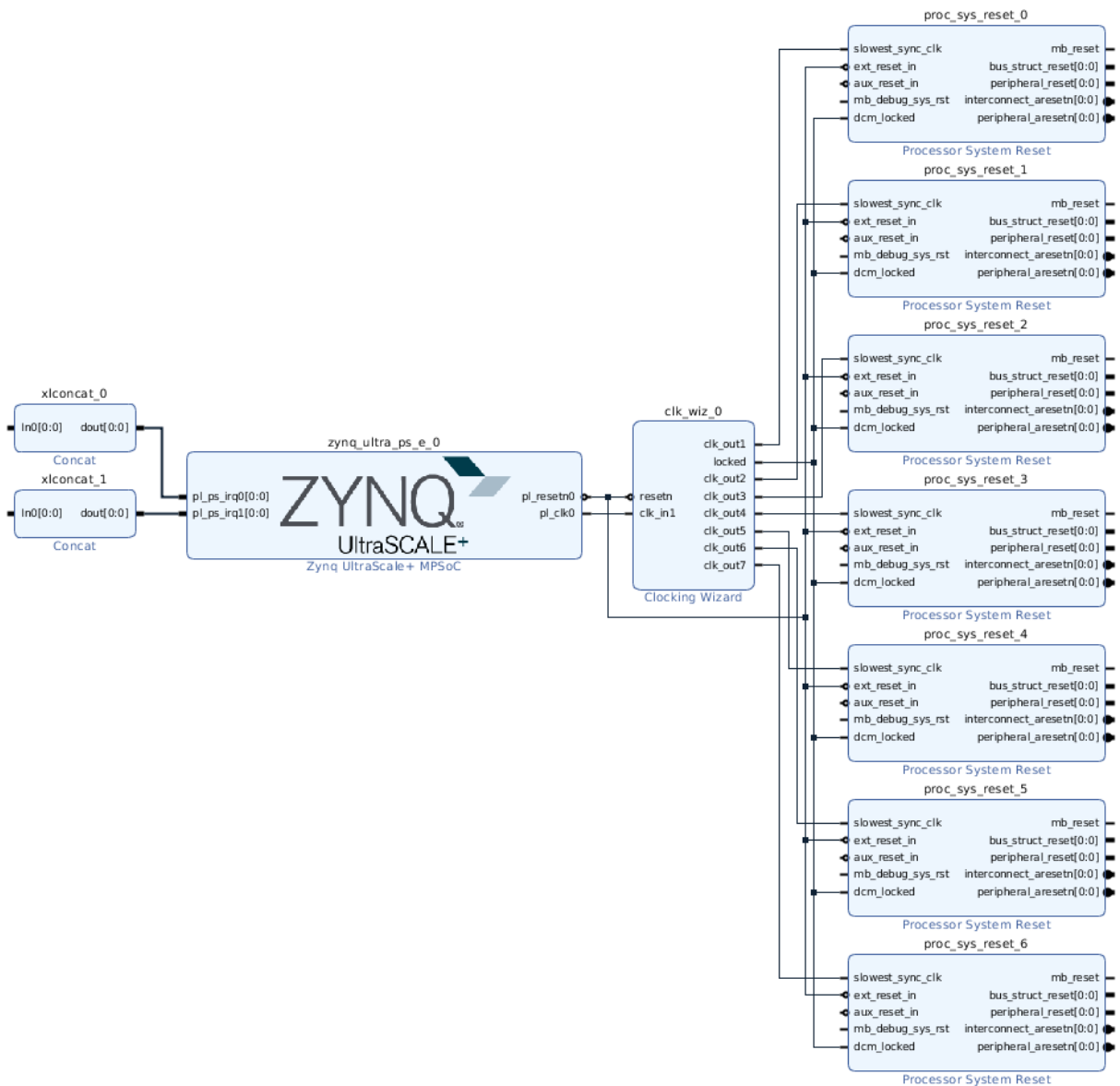
Αντιγράψτε το **Concat** IP έτσι ώστε να έχετε συνολικά 2.

Εκτελέστε το **Connection Automation** και αντιστοιχίστε στο κάθε σήμα **slowest_sync_clk** του κάθε καινούργιου **Processor System Reset** IP το κάθε καινούργιο ρολόι από το **Clocking Wizard** IP (π.χ. το **slowest_sync_clk** του **proc_sys_reset_6** αντιστοιχίζεται με το σήμα **clk_out7** του **clk_wiz_0**).

Κάντε τις εξής διασυνδέσεις “με το χέρι”:

- το σήμα εξόδου **pl_resethn0** του **Zynq UltraScale+ MPSoC** IP συνδέεται στα σήματα εισόδου **ext_reset_in** των **Processor System Reset** IPs
- το σήμα εξόδου **locked** του **Clocking Wizard** IP συνδέεται στα σήματα εισόδου **dcm_locked** των **Processor System Reset** IPs
- το σήμα εξόδου **dout[0:0]** του **xlconcat_1 Concat** IP συνδέεται στο σήμα εισόδου **pl_ps_irq1[0:0]** του **Zynq UltraScale+ MPSoC** IP

Κάντε **Regenerate Output** και **Validate Design**. Η σχεδίασή σας θα πρέπει να είναι όμοια με την πιο κάτω εικόνα (οι διασυνδέσεις πρέπει να είναι οι ίδιες παρά τη διαφορετική διαρρύθμιση των blocks).



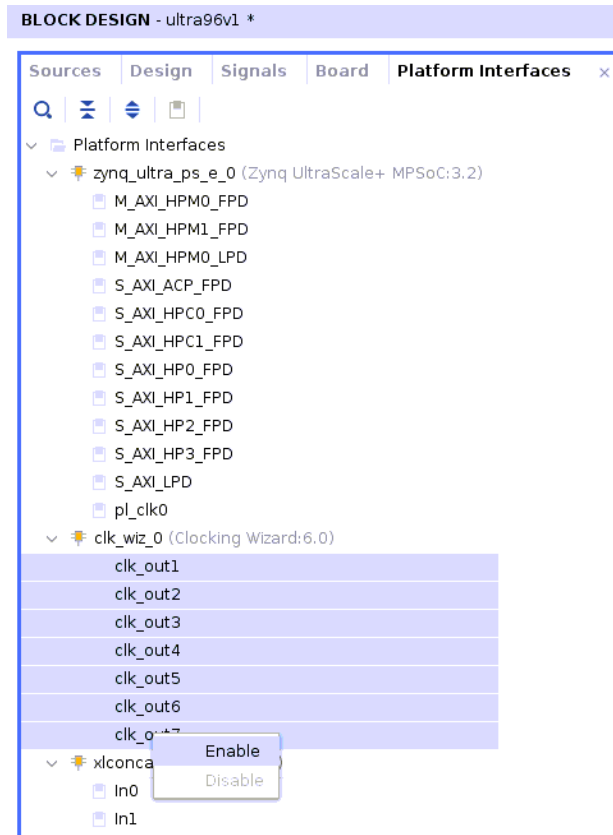
5. Καθορισμός των hardware διεπαφών

Σε αυτό το βήμα της σχεδίασης θα καθορίσουμε τις ιδιότητες της πλατφόρμας μας ώστε να μπορούν να χρησιμοποιηθούν από το εργαλείο SDSoC. Συγκεκριμένα, θα καθορίσουμε:

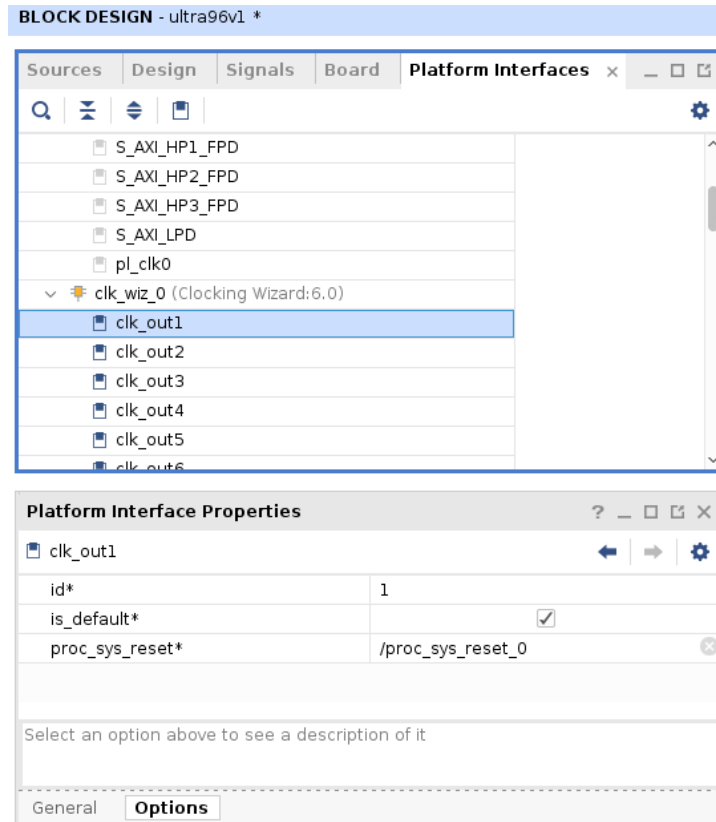
- τις διεπαφές για την επικοινωνία δεδομένων μεταξύ PS και PL
- τα διαθέσιμα σήματα ρολογιού
- τα διαθέσιμα σήματα διακοπών

Για το σκοπό αυτό, επιλέγουμε από το κυρίως μενού του Vivado **Window -> Platform Interfaces** και κάνουμε κλικ πάνω στο **Enable platform interfaces**.

Το τμήμα **Platform Interfaces** εμφανίζει όλα τα σήματα της σχεδίασής μας που θα μπορούσαν να επιλεγούν για να γίνουν διαθέσιμα στην πλατφόρμα μας. Μπορούμε να επιλέξουμε ποια σήματα θα γίνουν διαθέσιμα στην πλατφόρμα επιλέγοντας τα επιθυμητά σήματα, πατώντας **δεξί κλικ** και επιλέγοντας το **Enable**.



Κάτω από το τμήμα **Platform Interfaces** υπάρχει το τμήμα **Platform Interface Properties** στο οποίο μπορούμε να καθορίσουμε ποιο θα είναι το προεπιλεγμένο (default) σήμα ρολογιού κατά τη χρήση της πλατφόρμας από το SDSoC. Επιλέγουμε το **clk_out1** (100MHz) και στο τμήμα **Platform Interface Properties** επιλέγουμε το **is_default***.



Με τον ίδιο τρόπο επιλέξτε και ενεργοποιήστε τα AXI interfaces για επικοινωνία δεδομένων μεταξύ PS και PL - ενεργοποιήστε τα όλα εκτός από τα **pl_clk0**, **S_AXI_ACP_FPD**, και **S_AXI_LPD**.

Τέλος, ενεργοποιήστε τα σήματα διακοπής από 0 μέχρι και 7 για τα **Concat** IPs.

6. Παραγωγή HDL και DSA αρχείων

Σε αυτό το βήμα θα δημιουργήσουμε HDL αρχεία για τη σχεδίασή μας. Στο τμήμα **Sources** βρίσκουμε το **ultra96v1 (ultra96v1.bd)** πατάμε **δεξί κλικ** πάνω του και επιλέγουμε **Generate Output Products**. Στο παράθυρο που θα ανοίξει επιλέγουμε **Generate**.

The following output products will be generated.



Preview

Search icons: [magnifying glass], [expand], [collapse]

▼ ultra96v1.bd (00C per IP)

- Synthesis
- Implementation
- Simulation

Synthesis Options

☐ Global

☒ Out of context per IP

☐ Out of context per Block Design

Run Settings

☒ On local host: Number of jobs: 8 ▼

☐ On remote hosts:

☐ Use LSF:

Αφού τελειώσει η παραπάνω ενέργεια (και το παράθυρο του Vivado γράφει Ready πάνω δεξιά), πατάμε ξανά **δεξί κλικ** πάνω στο **ultra96v1 (ultra96v1.bd)** και επιλέγουμε **Create HDL Wrapper** για τη δημιουργία HDL περιγραφής της πλατφόρμας μας. Επιλέγουμε **OK**.

You can either add or copy the HDL wrapper file to the project. Use copy option if you would like to modify this file.



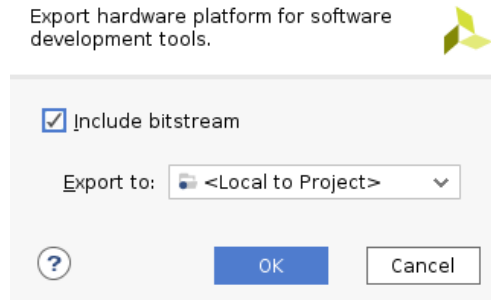
Options

☐ Copy generated wrapper to allow user edits

☒ Let Vivado manage wrapper and auto-update

Στη συνέχεια, επιλέγουμε τη δημιουργία του bitstream αρχείου της μέχρι τώρα σχεδίασης. Με αυτό το αρχείο θα μπορούσαμε να προγραμματίσουμε το FPGA για να επιτελέσει κάποια λειτουργία (προς το παρόν μας υπολείπεται ο επιταχυντής οπότε δεν θα το χρησιμοποιήσουμε). Για το σκοπό αυτό, επιλέγουμε στο **Flow Navigator** το **PROGRAM AND DEBUG -> Generate Bitstream**. Επιλέγουμε **Yes** για να εκτελεστούν τα **synthesis and implementation** και στο επόμενο παράθυρο επιλέγουμε **OK**.

Μετά το τέλος της προηγούμενης ενέργειας, επιλέγουμε **Cancel** στο παράθυρο που θα ανοίξει. Στη συνέχεια, επιλέγουμε **File -> Export > Export Hardware** από το κυρίως μενού και επιλέγουμε το **Include Bitstream** για να κάνουμε εξαγωγή της μέχρι τώρα σχεδίασής μας.



Τέλος, εκτελούμε τις εξής 2 εντολές στο **Tcl Console** (που βρίσκεται στο κάτω μέρος του παραθύρου του Vivado) για τη δημιουργία και την επαλήθευση του DSA αρχείου.

```
write_dsa -force -include_bit <τοποθεσία_αρχείου>/όνομα_αρχείου.dsa
```

```
validate_dsa <τοποθεσία_αρχείου>/όνομα_αρχείου.dsa
```

π.χ.

```
write_dsa -force -include_bit  
/mnt/terabyte/pmousoul_data/projects/vivado_2018.2/ultra96v1/ultra96v1.dsa
```

```
validate_dsa  
/mnt/terabyte/pmousoul_data/projects/vivado_2018.2/ultra96v1/ultra96v1.dsa
```

Μπορείτε να κλείσετε το Vivado. Το παραγόμενο DSA αρχείο αναπαριστά το υλικό μέρος της πλατφόρμας. Στο Εργαστήριο 3 θα δημιουργήσουμε το λογισμικό μέρος της πλατφόρμας με τη χρήση του PetaLinux και τελικά την πλατφόρμα με τη χρήση του εργαλείου SDSoc. Στη συνέχεια, θα δώσουμε στο SDSoc ως είσοδο τα παραγόμενα αρχεία που αναπαριστούν το λογισμικό συστήματος και το υλικό της πλατφόρμας. Τέλος, θα χρησιμοποιήσουμε αυτή την πλατφόρμα για να αναπτύξουμε την εφαρμογή μας.