



---

**ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ**  
**ΣΧΟΛΗ Η.Μ.Μ.Υ.**  
**ΕΡΓΑΣΤΗΡΙΟ ΜΙΚΡΟΕΠΕΞΕΡΓΑΣΤΩΝ & ΥΛΙΚΟΥ**  
**ΗΡΥ591 ΑΝΑΔΙΑΤΑΣΣΟΜΕΝΑ ΨΗΦΙΑΚΑ ΣΥΣΤΗΜΑΤΑ**

---

**Αναφορά milestone 3**

<b><u>Ομάδα Εργασίας:</u></b>	<b>LAB59140120</b>
<b>Μποκαλίδης Αναστάσιος</b>	<b>2014030069</b>
<b>Χατζηπέτρος Αλέξανδρος</b>	<b>2013030151</b>

### **Σκοπός 3<sup>ης</sup> εργαστηριακής άσκησης**

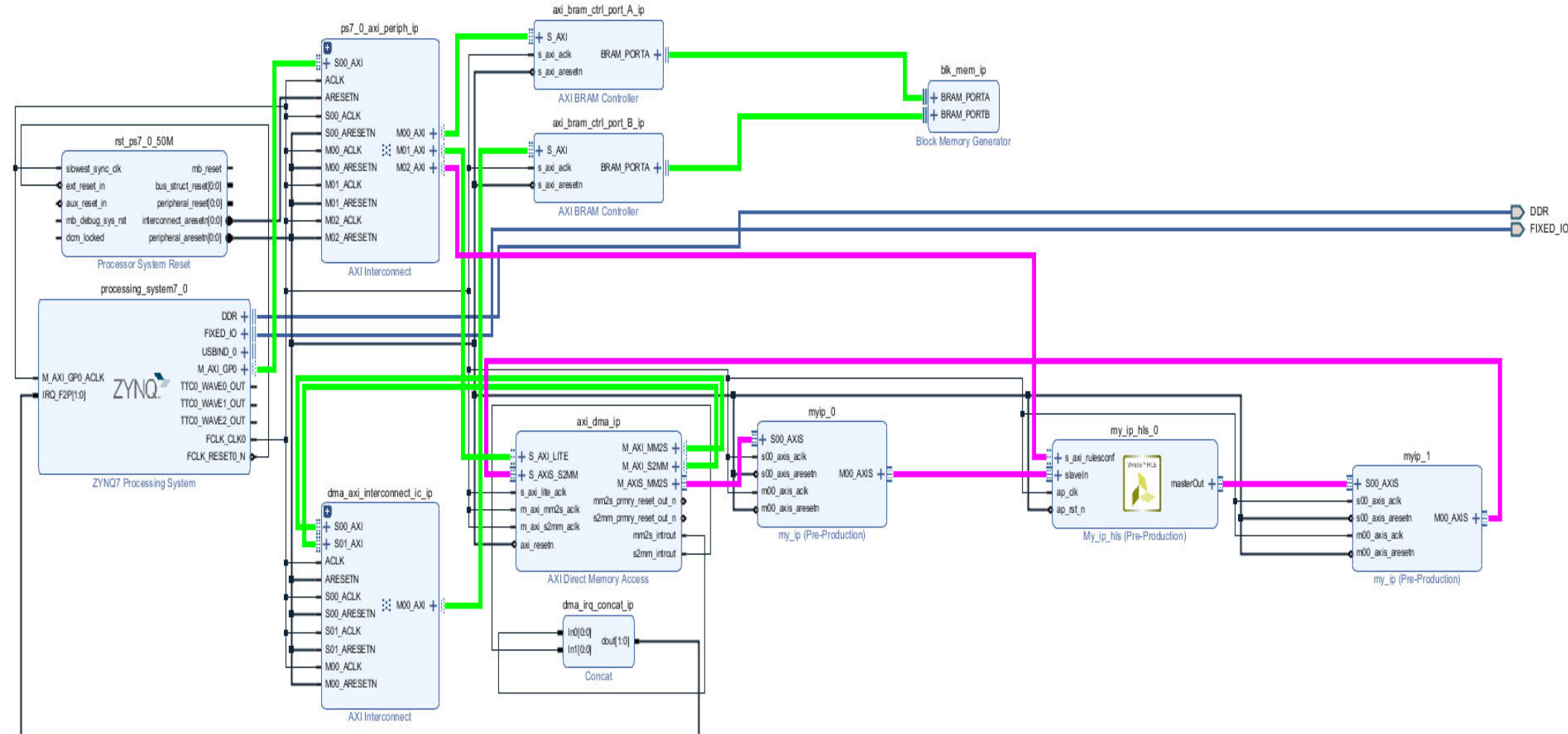
Σκοπός της άσκησης αυτής είναι να ενσωματώσουμε τα προηγούμενα milestones στο συνολικό μας σύστημα και να δούμε όλη την ροή δεδομένων, από την δημιουργία στο φιλτράρισμα και πάλι στην αποθήκευση της στην μνήμη.

### **Περιγραφή εργαστηριακής άσκησης-Κυματομορφές**

Παρακάτω επεξηγούνται τα βήματα για την ολοκλήρωση της άσκησης αυτής :

- 1) Αρχικά εφόσον είχαμε υλοποιήσει το milestone 2 στο Vivado HLS επιτυχώς το μόνο μόνο που χρειάστηκε ήταν να κάνουμε ένα τελικό simulation και έπειτα export του RTL σχεδίου ώστε να μπορέσουμε να το χρησιμοποιήσουμε ως ip-block στο τελικό σύστημα. Αυτό έγινε σύμφωνα με τα βήματα που δίνονται στην εκφώνηση. Επίσης κατά την αρχικοποίηση των κανόνων rule0, rule1, rule2 που κάναμε στο testbench του Vivado βάλαμε τις τιμές 10,30,50 (σε δεκαεξαδικό) αντίστοιχα.
- 2) Έπειτα πάλι με βάση την εκφώνηση της 3<sup>ης</sup> άσκησης γυρίσαμε στο Vivado και ενσωματώσαμε το my\_ip\_hls κατάλληλα στο τελικό μας σχέδιο. Εκεί το ενώσαμε με 2 instances του myip που φτιάξαμε στο milestone 1. Δηλαδή η έξοδος master του myip\_0 συνδέεται με την είσοδο slave του my\_ip\_hls\_0 και η έξοδος master του my\_ip\_hls\_0 συνδέεται με την είσοδο slave του myip\_1. Όσον αφορά το s\_axi\_rulesConf χρησιμοποιήσαμε την αυτόματη σύνδεση και συνδέθηκε με το κατάλληλο block του σχεδίου το ps7\_0\_axi\_periph\_ip. Εδώ πρέπει να αναφέρουμε χρειάστηκε να κάνουμε μια μικρή αλλαγή σε 1 σήμα στο myip του 1<sup>ου</sup> milestone έτσι ώστε να μπορέσει να συγχρονιστεί καλύτερα με τα υπόλοιπα blocks.

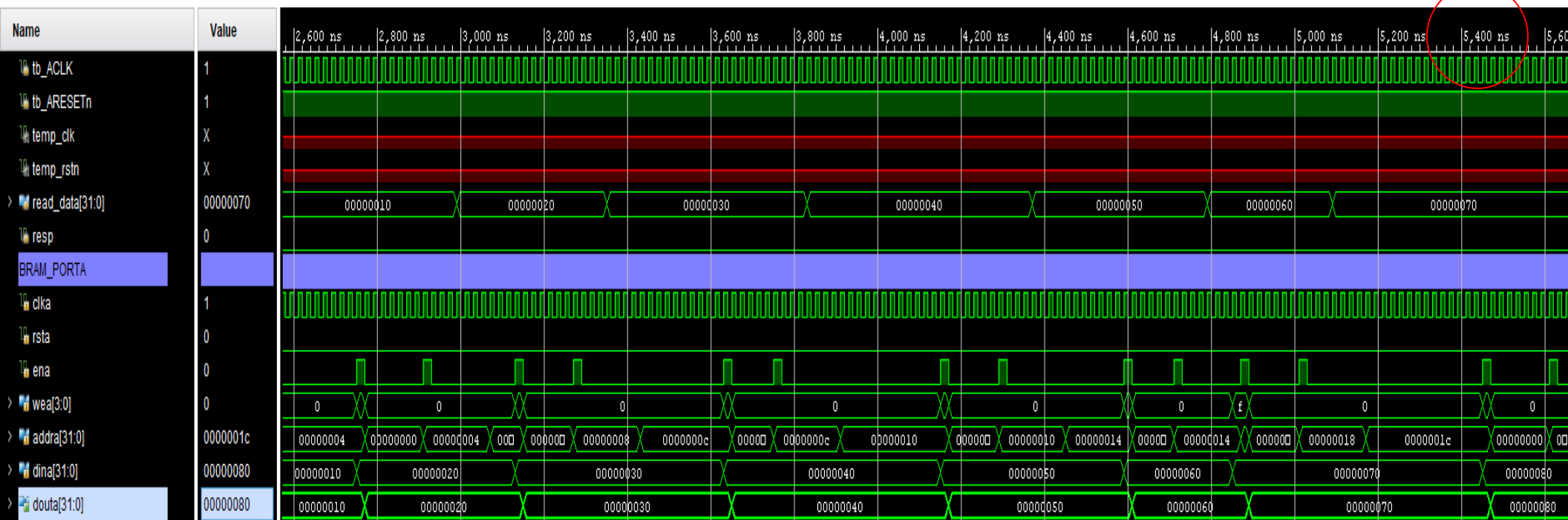
3) Ακολουθεί το **Block Diagram** του συνολικού συστήματος : Με πράσινο και ροζ χρώμα είναι οι απαραίτητες διαδρομές που καθορίζουν την ροή δεδομένων μας και την κίνηση τους στο συνολικό σύστημα. Με ροζ χρώμα είναι οι διαδρομές από/προς τα έξτρα modules που έχουμε υλοποιήσει στα milestone 1 και 2.



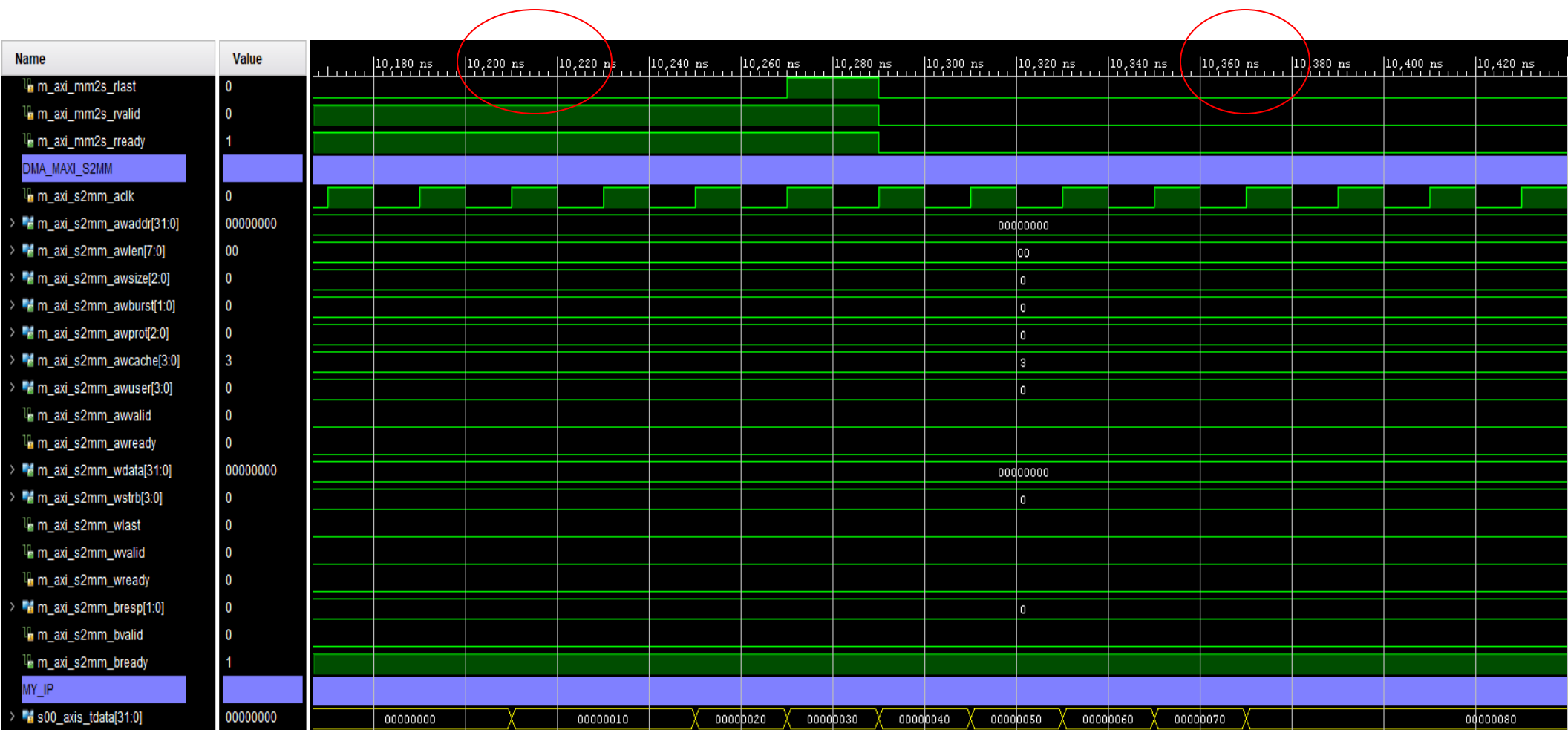
4) Από το παραπάνω σχήμα μπορούμε να καταλάβουμε και την ροή των δεδομένων μας (10,20,30,40,50,60,70,80 δεκαεξαδικής μορφής).

- I. Αρχικά ο **επεξεργαστής** γράφει τα δεδομένα προς επεξεργασία στην **BRAM PORTA** και προγραμματίζει τον **DMA** ώστε να ξέρει που να αποθηκεύσει τα δεδομένα του καθώς και τον αριθμό των δεδομένων αυτών.
- II. Έπειτα ο **επεξεργαστής** ενημερώνει πάλι τον **DMA** από που να πάρει τα δεδομένα που στη συνέχεια θα σταλθούν στην λογική που έχουμε προσθέσει.
- III. Τα δεδομένα από τον **DMA** μεταφέρονται και αποθηκεύονται στην πρώτη μας **FIFO**, δηλαδή στο **myip\_0**. Αναφορικά εισέρχονται στο slave interface του module.
- IV. Συνεχίζοντας τα δεδομένα που βγαίνουν από το master interface του **myip\_0** εισέρχονται στο **my\_ip\_hls\_0** και συγκεκριμένα στο slave interface του. Εκεί η ροή των δεδομένων μας φιλτράρεται μέσω των κανόνων μας. Δηλαδή θα εισέλθει η ροή 10,20,30,40,50,60,70,80 , έπειτα θα φιλτραριστεί από τους κανόνες 10,30,50 και η τελική ροή θα είναι 20,40,60,70,80.
- V. Όταν φιλτραριστεί η ροή δεδομένων μας είναι έτοιμη να περάσει στην δεύτερη FIFO μας, δηλαδή στο **myip\_1**. Εισέρχεται στο slave interface και μετά από το master interface επιστρέφει πίσω στο **DMA**.
- VI. Τέλος αφού η ροή δεδομένων περάσει από την λογική μας και υποστεί την κατάλληλη επεξεργασία, μέσω του **DMA** γράφονται πίσω στην **BRAM PORTB**.

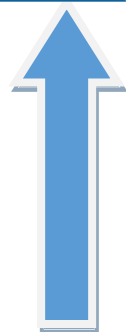
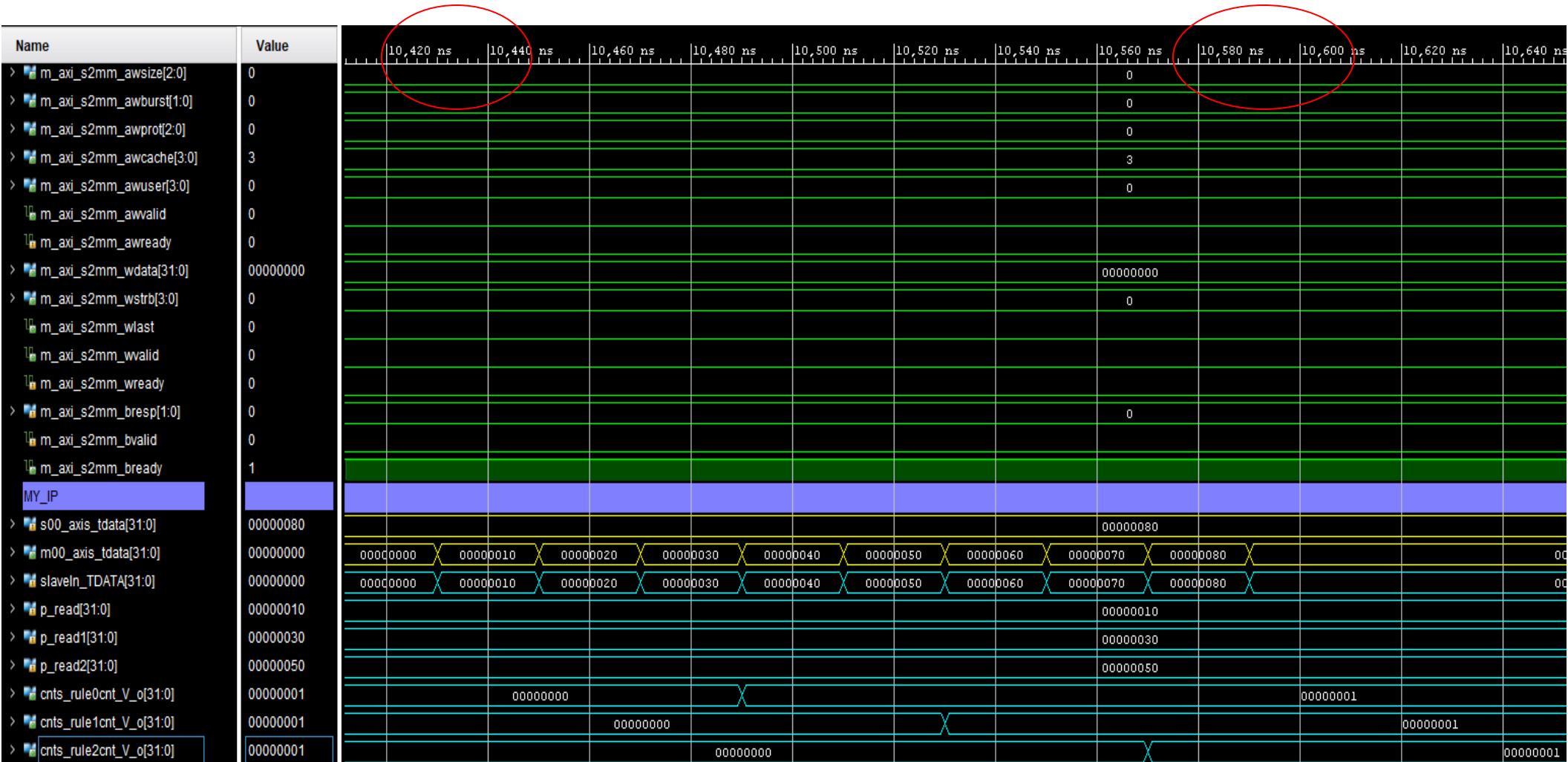
5) Παρακάτω παρατίθενται εικόνες-επεξηγήσεις από το simulation του συνολικού συστήματος, με κόκκινο κύκλο απεικονίζονται οι χρονικές στιγμές για να φαίνεται καλύτερα η ροή δεδομένων από block σε block.



Εδώ βλέπουμε την στιγμή που εισέρχονται τα δεδομένα στην μνήμη **BRAM** για πρώτη φορά και τότε εξέρχονται για πρώτη φορά (dina[31:0] και douta[31:0] αντίστοιχα).



Εδώ βλέπουμε τις στιγμές όπου τα δεδομένα εισέρχονται στο **myip\_0**. Με κίτρινο σήμα είναι η είσοδος του module αυτού(FIFO\_0).



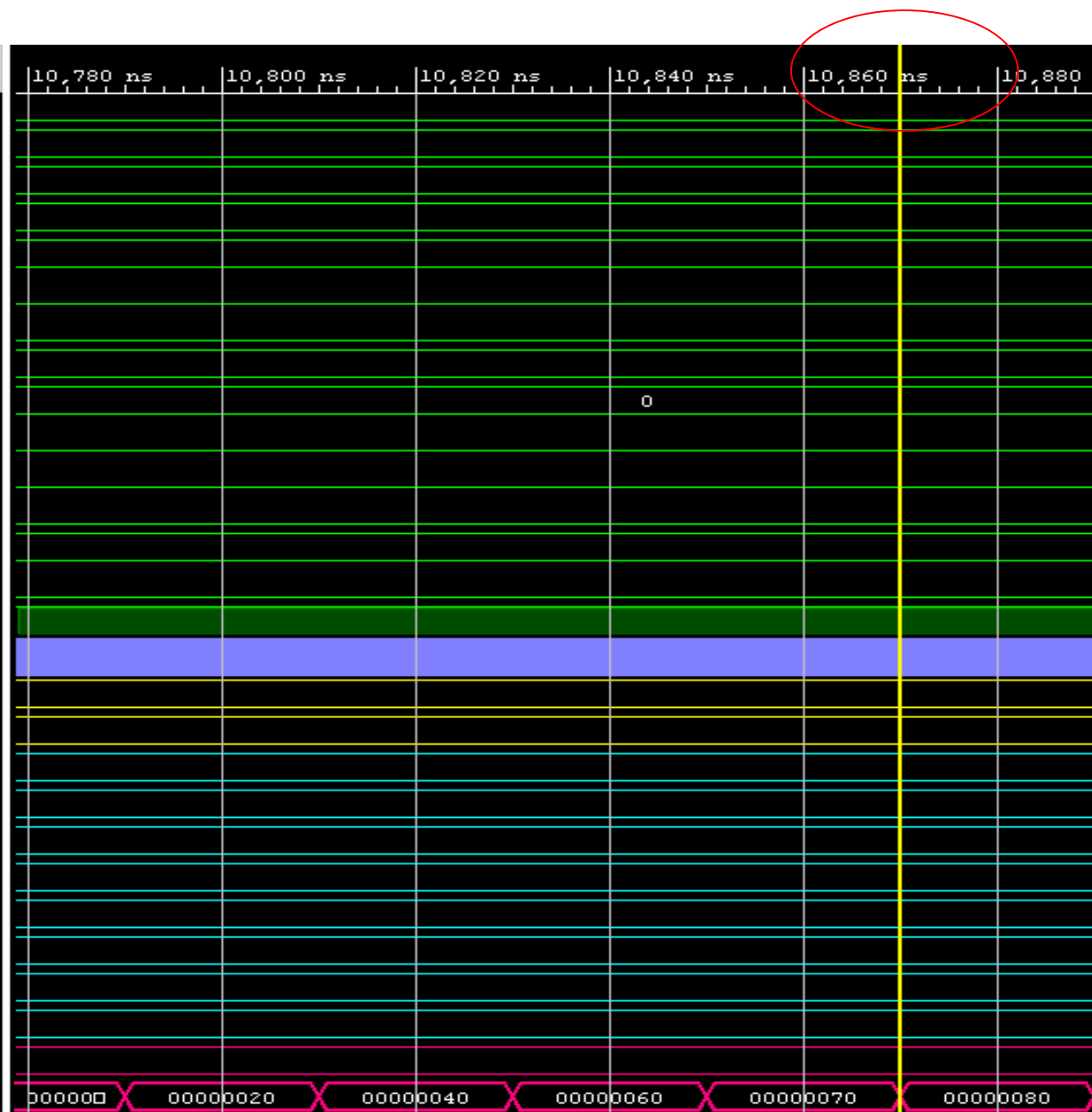
Εδώ βλέπουμε τις στιγμές που τα δεδομένα βγαίνουν από το myip\_0 και εισέρχονται στο my\_ip\_hls\_0. Επίσης φαίνονται και οι τιμές που παίρνουν οι καταχωρητές των κανόνων καθώς και οι τιμές των μετρητών που κρατάνε το πόσες φορές πετύχαμε ένα κανόνα. Με μπλε χρώμα είναι τα απαραίτητα σήματα του my\_ip\_hls\_0 : 1<sup>ο</sup> σήμα η είσοδος, 2<sup>ο</sup> σήμα κανόνας 1<sup>ος</sup> , 3<sup>ο</sup> σήμα κανόνας 2<sup>ος</sup> , 4<sup>ο</sup> σήμα κανόνας 3<sup>ος</sup> , 5<sup>ο</sup> σήμα μετρητής 1<sup>ος</sup> , 6<sup>ο</sup> σήμα μετρητής 2<sup>ος</sup> , 7<sup>ο</sup> σήμα μετρητής 3<sup>ος</sup> .

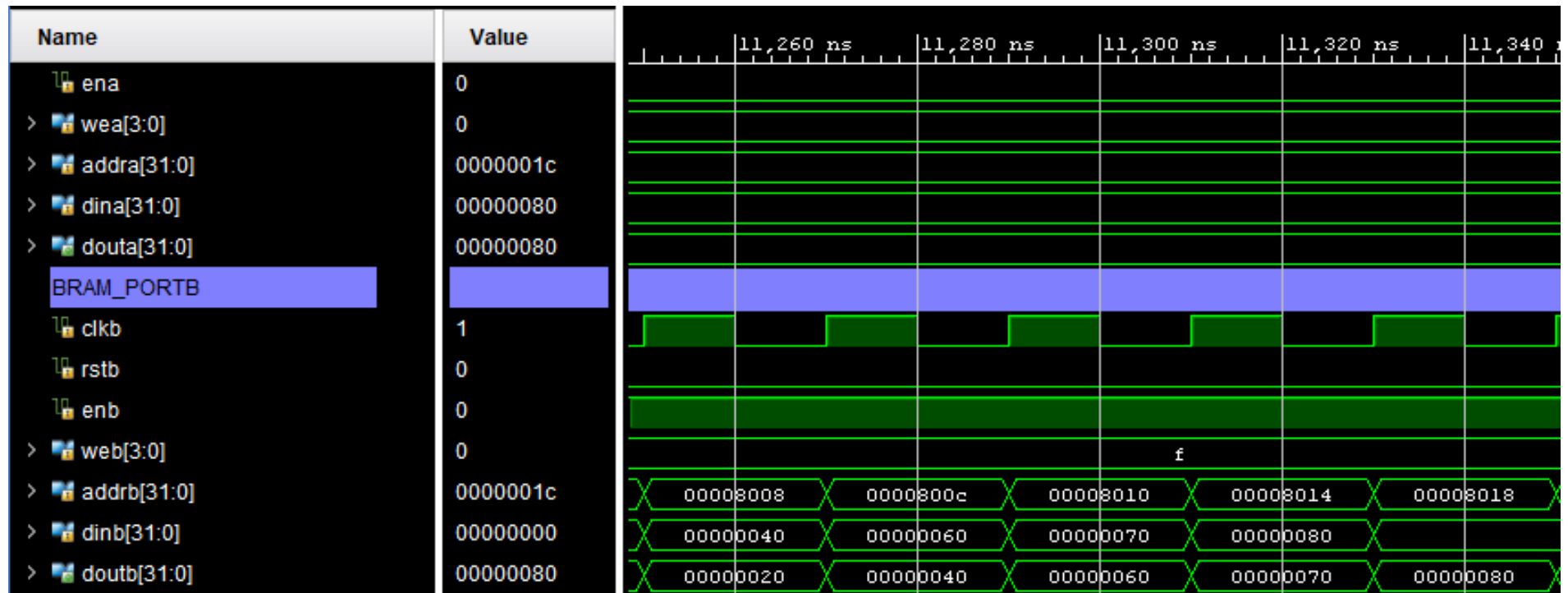


Εδώ βλέπουμε την χρονική στιγμή όπου τα φιλτραρισμένα δεδομένα (20,40,60,70,80) περνάνε από το **my\_ip\_hls\_0** στο **myip\_1** (2<sup>η</sup> FIFO με ροζ χρώμα) καθώς και την στιγμή που φεύγουν από το **myip\_1**.



Name	Value
>  m_axi_s2mm_awsiz[2:0]	0
>  m_axi_s2mm_awburst[1:0]	0
>  m_axi_s2mm_awprot[2:0]	0
>  m_axi_s2mm_awcache[3:0]	3
>  m_axi_s2mm_awuser[3:0]	0
m_axi_s2mm_awvalid	0
m_axi_s2mm_awready	0
>  m_axi_s2mm_wdata[31:0]	00000000
>  m_axi_s2mm_wstrb[3:0]	0
m_axi_s2mm_wlast	0
m_axi_s2mm_wvalid	0
m_axi_s2mm_wready	0
>  m_axi_s2mm_bresp[1:0]	0
m_axi_s2mm_bvalid	0
m_axi_s2mm_bready	1
MY_IP	
>  s00_axis_tdata[31:0]	00000080
>  m00_axis_tdata[31:0]	00000000
>  slaveIn_TDATA[31:0]	00000000
>  p_read[31:0]	00000010
>  p_read1[31:0]	00000030
>  p_read2[31:0]	00000050
>  cnts_rule0cnt_V_o[31:0]	00000001
>  cnts_rule1cnt_V_o[31:0]	00000001
>  cnts_rule2cnt_V_o[31:0]	00000001
>  masterOut_TDATA[31:0]	00000000
>  s00_axis_tdata[31:0]	00000000
>  m00_axis_tdata[31:0]	00000080





Τέλος βλέπουμε τις χρονικές στιγμές όπου τα φιλτραρισμένα δεδομένα επιστρέφουν στην μνήμη **BRAM PORTB**.

