

Άσκηση 1: Performance and resources measurement

A) Τα αποτελέσματα του *Estimate Performance* χωρίς κανένα optimization είναι:

Details

Performance estimates for 'forward_propagation in main.cp ...

HW accelerated (Estimated cycles) 683780

Resource utilization estimates for HW functions

Resource	Used	Total	% Utilization
DSP	3	80	3,75
BRAM	16	60	26,67
LUT	1760	17600	10
FF	892	35200	2,53

B) Τρέχοντας την εφαρμογή στο Zybo board, διαπιστώνουμε τα εξής:

Η εφαρμογή απαιτεί **682935 κύκλους** για την εκτέλεσή της, αριθμός που **συμφωνεί** με το *estimation* του ερωτήματος (A).

Το speed-up σε σχέση με την SW εκτέλεση στον ARM είναι **2.16051**

```
sh-4.3# ./ask4.elf
Starting dataset parsing...
Parsing finished...
Starting hardware calculations...
Hardware calculations finished.
Starting software calculations...
Software calculations finished.
Hardware cycles : 682935
Software cycles : 1475490
Speed-Up       : 2.16051
Saving results to output.txt...
```

Γ) Δοκιμάσαμε τα παρακάτω optimizations:

#pragma HLS pipeline II=1

Details

Performance estimates for 'forward_propagation in main.cp ...

HW accelerated (Estimated cycles) 86190

Resource utilization estimates for HW functions

Resource	Used	Total	% Utilization
DSP	80	80	100
BRAM	30	60	50
LUT	3737	17600	21,23
FF	5811	35200	16,51

→ Σημαντική μείωση των κύκλων, αλλά μέγιστη κατανάλωση των DSPs και κατασπατάληση των πόρων.

4η Άσκηση – Σχεδιασμός Ενσωματωμένων Συστημάτων

```
#pragma HLS array_partition variable=xbuf complete
#pragma HLS array_partition variable=layer_1_out complete
```

Details

Performance estimates for 'forward_propagation in main.cp ...

HW accelerated (Estimated cycles) 86092

Resource utilization estimates for HW functions

Resource	Used	Total	% Utilization
DSP	80	80	100
BRAM	30	60	50
LUT	4589	17600	26,07
FF	10789	35200	30,65

→ Παρατηρούμε διπλασιασμό της χρήσης των **FFs**. Διαθέτουμε, όμως, σημαντικό περιθώριο (30% είναι η συνολική χρήση), οπότε η αύξηση αυτή δεν επηρεάζει την εκτέλεση της εφαρμογής. Ωστόσο, για μια τόσο αμελητέα βελτίωση στην απόδοση (**μείωση των κύκλων κατά 0.12%**), ο διπλασιασμός της χρήσης πόρων ενδέχεται να οδηγήσει σε αισθητή αύξηση της κατανάλωσης ισχύος.

```
#pragma HLS unroll factor=<2, 30, 50>
```

Details

Performance estimates for 'forward_propagation in main.cp ...

HW accelerated (Estimated cycles) 10718

Resource utilization estimates for HW functions

Resource	Used	Total	% Utilization
DSP	80	80	100
BRAM	40	60	66,67
LUT	10869	17600	61,76
FF	11991	35200	34,07

→ Με την προσθήκη των loop-unrolling HLS directives, παρατηρούμε μία **σημαντική μείωση των κύκλων** που απαιτούνται, καθώς πολλές από τις πράξεις εκτελούνται πλέον παράλληλα. Παρά το γεγονός ότι το utilization των πόρων **αυξάνεται δραματικά** (για **BRAM** και **LUTs**), εξακολουθεί να παραμένει σε **αποδεκτές τιμές**.

*Μεγαλύτερες τιμές factor **δεν επηρεάζαν** την επίδοση της εφαρμογής ή/και **χειροτέρευαν την χρησιμοποίηση** των πόρων ακόμα και **πάνω από τα επιτρεπτά επίπεδα**.

Η τελική optimized υλοποίηση της εφαρμογής βρίσκεται στο επισυναπτόμενο αρχείο **network.cpp**.

```
sh-4.3# ./ask4.elf
Starting dataset parsing...
Parsing finished...
Starting hardware calculations...
Hardware calculations finished.
Starting software calculations...
Software calculations finished.
Hardware cycles : 10906
Software cycles : 1476495
Speed-Up       : 135.384
Saved results to output.txt...
```

Τρέχοντας την εφαρμογή στο Zybo board, διαπιστώνουμε ότι η εφαρμογή απαιτεί **10906 κύκλους** για την εκτέλεσή της, αριθμός που συμφωνεί με το παραπάνω estimation, ενώ το speed-up σε σχέση με την SW εκτέλεση στον ARM είναι **135.384**. Η βελτίωση που παρουσιάζει σε σχέση με την unoptimized εκδοχή όσο αφορά τον χρόνο εκτέλεσης είναι παραπάνω από σημαντική, ωστόσο απαιτεί περισσότερους πόρους για να υλοποιηθεί.

4η Άσκηση – Σχεδιασμός Ενσωματωμένων Συστημάτων

 $\Delta)$

Latency (clock cycles)

Summary

LatencyInterval

minmaxminmaxType

1077107710781078none

Detail

Instance

Loop

	Latency		Initiation Interval		Trip Count		Pipelined
Loop Name	min	max	Iteration Latency	achievedtarget			
- read_input	395	395	4	11	392		yes
- layer_1	197	197	3	11	196		yes
- layer_2	52	52	4	11	50		yes
- layer_3	400	400	10	11	392		yes

→ Το **layer_3** παρουσιάζει το μεγαλύτερο latency (**400 κύκλοι**). Η optimized υλοποίηση μας είναι fully pipelined (**initiation interval achieved = 1** για όλα τα loops)

Resource Profile

	BRAM	DSP	FF	LUT	Bits P0	Bits P1	Bits P2	Banks/Depth	Words	W*Bits*Banks
forward_propagation	81	80	11991	10785						
I/O Ports(2)				64						
Instances(2)	0	0	268	677						
Memories(112)	81		305	245	1022			112	33934	307474
Expressions(1085)	0	80	0	7326	7213	8555	3106			
-	0	0	0	78	16	78	0			
*	0	80	0	0	1145	956	0			
+	0	0	0	3054	3541	3530	0			
and	0	0	0	6	6	6	0			
ashr	0	0	0	161	54	54	0			
icmp	0	0	0	660	1955	1565	0			
or	0	0	0	210	206	198	0			
select	0	0	0	3117	267	2141	3106			
shl	0	0	0	19	10	10	0			
xor	0	0	0	21	13	17	0			
Registers(1120)			11418		12187					
Channels(0)	0		0	0	0			0	0	0
Multiplexers(137)	0		0	2537	1776			0		

→ Η έκφραση “*” απαιτεί τα περισσότερα DSPs: **80** (όσα δηλαδή είναι διαθέσιμα).

4η Άσκηση – Σχεδιασμός Ενσωματωμένων Συστημάτων

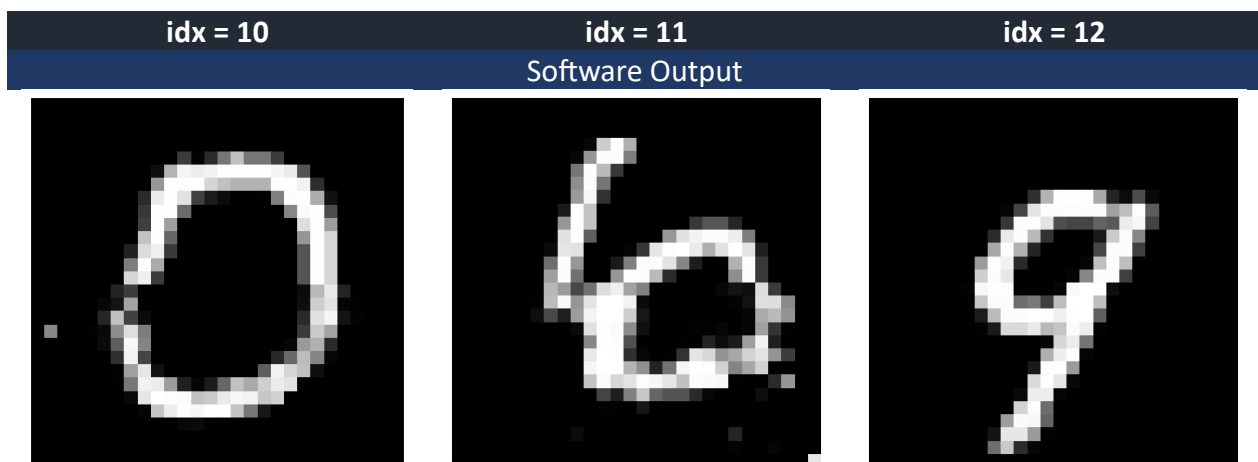
Άσκηση 2: Quality Measurement

A) Τρέχοντας το `plot_output.ipynb` σε συνδυασμό με το `output.txt` που παρήγαγε το εκτελέσιμο, λαμβάνουμε τις εξής combined εικόνες από SW και HW για `idx: 10, 11, 12`.

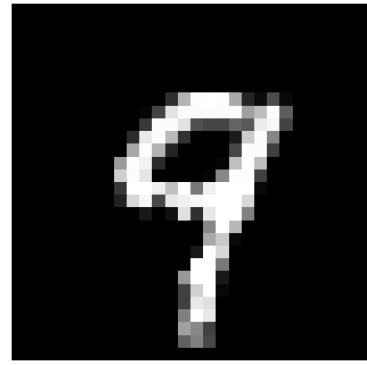
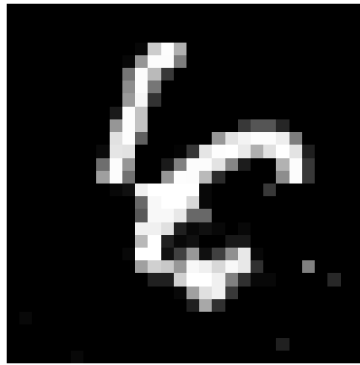
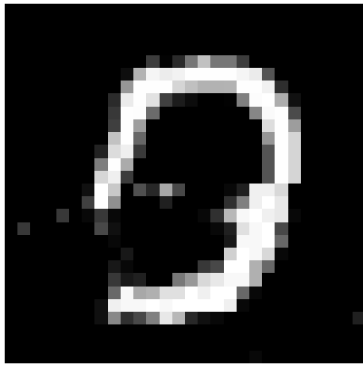


B) Αλλάζοντας τα ορίσματα `BITS` και `BITS_EXP` στο αρχείο `network.h`, παράγουμε designs με νέα δεκαδική ακρίβεια στα datatypes που χρησιμοποιούμε, συγκεκριμένα 4-bit και 10-bit (στο προηγούμενο ερώτημα είχαμε 8-bit δεκαδική ακρίβεια). Στην συνέχεια, παράγουμε τις combined εικόνες για `idx: 10, 11, 12` και για τα 2 designs.

- 4-bit ακρίβεια: Για ακρίβεια 4 bits, παρατηρούμε ότι οι εικόνες ειδικά για το “0” και το “6” δεν έχουν την μορφή που περιμέναμε, πράγμα που σημαίνει ότι η εφαρμογή δεν είναι λειτουργική.



Hardware Output



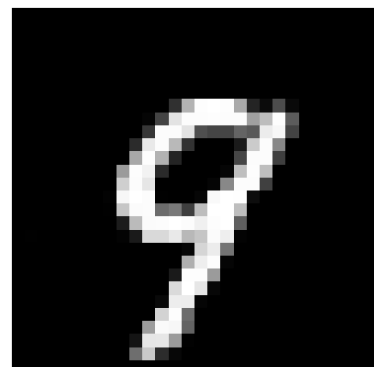
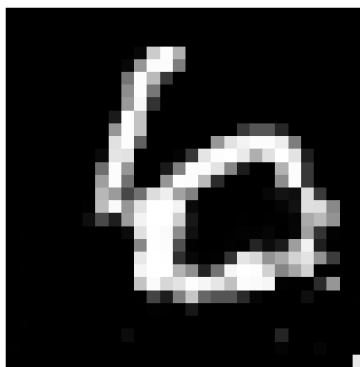
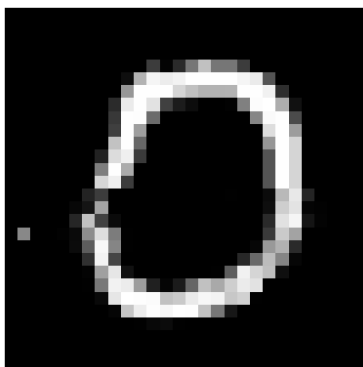
- 10-bit ακρίβεια: Παρατηρούμε ότι για 10-bit δεκαδική ακρίβεια οι combined εικόνες που παράγονται δεν εμφανίζουν αισθητές διαφορές σε σχέση με τις αντίστοιχες αρχικές (8-bit ακρίβειας design).

idx = 10

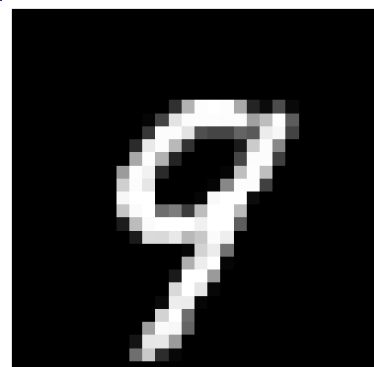
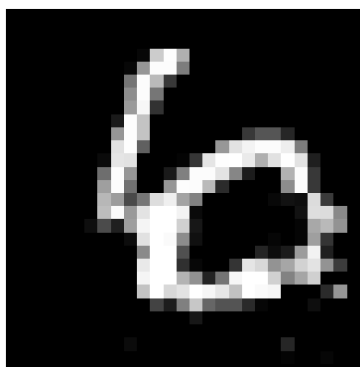
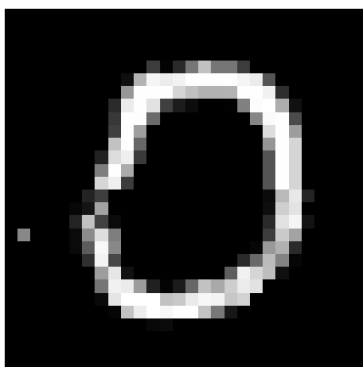
idx = 11

idx = 12

Software Output



Hardware Output



Γ) Παρακάτω παρουσιάζονται τα **Max pixel errors** και τα **Peak Signal-to-Noise Ratio** για τα διάφορα bits που δοκιμάσαμε.

- Ακρίβεια 4 bits

Index	10	11	12
Max pixel error	255	249	255
Peak Signal-to-Noise Ratio	14.051663945384881	14.634988266184102	13.525831164368576

4η Άσκηση – Σχεδιασμός Ενσωματωμένων Συστημάτων

- Ακρίβεια 8 bits

Index	10	11	12
Max pixel error	16	17	13
Peak Signal-to-Noise Ratio	42.6822168370888	42.56993337396983	47.065287020211215

- Ακρίβεια 10 bits

Index	10	11	12
Max pixel error	5	5	4
Peak Signal-to-Noise Ratio	54.08543347142643	52.556099880280584	53.76982650203158

Αξιολόγηση Μετρικών

Ανάμεσα στις δύο μετρικές προτιμούμε την μετρική **Peak Signal-to-Noise Ratio (PSNR)**, καθώς εκφράζει τη συνολική ποιότητα της ανακατασκευασμένης εικόνας λαμβάνοντας υπόψη τη σχέση μεταξύ σήματος και θορύβου. Αντίθετα, η μετρική **Max Pixel Error** δεν θεωρείται τόσο αξιόπιστη, καθώς επηρεάζεται από μεμονωμένα pixels με μεγάλες αποκλίσεις, που όμως μπορεί να μην επηρεάζουν ουσιαστικά τη συνολική κατανομή των pixels.

Ποιότητα εικόνας ανάλογα με την ακρίβεια των bits

- ✓ **Ακρίβεια 10 bits:**
Εξασφαλίζει την καλύτερη ανακατασκευή εικόνας, με υψηλό SNR και πολύ μικρό Max Pixel Error, προσφέροντας αποτέλεσμα σχεδόν ταυτόσημο με την αρχική εικόνα.
- ✓ **Ακρίβεια 4 bits:**
Οδηγεί σε χαμηλή ποιότητα εικόνας, με έντονο θόρυβο (SNR = 14) και μεγάλη απόκλιση ανά pixel (Max Pixel Error = 255).
- ✓ **Ακρίβεια 8 bits:**
Παρέχει ποιότητα κοντά στα 10 bits, με μικρή διαφορά στο SNR και αποδεκτά επίπεδα Max Pixel Error.

Trade-off:

- ✓ **Μέγιστη Ποιότητα Ανακατασκευής:** Τα 10 bits παρέχουν την υψηλότερη ποιότητα ανακατασκευής, αλλά απαιτούν αυξημένους υπολογιστικούς πόρους. Αυτό επηρεάζει την απόδοση της εφαρμογής, καθώς η απαίτηση χρήσης περισσότερων πόρων από αυτούς που προσφέρονται μας οδηγεί στο να περιορίσουμε την εφαρμογή τεχνικών βελτιστοποίησης.

Ή

- ✓ **Ισορροπία Ποιότητας-Πόρων:** Εάν η απόδοση του συστήματος αποτελεί σημαντικό παράγοντα, η επιλογή bits ακρίβειας γύρω στην τιμή 8 προσφέρει μια ικανοποιητική ισορροπία. Διατηρείται καλή ποιότητα ανακατασκευής με σημαντικά χαμηλότερες απαιτήσεις σε πόρους.