Πλήρη Σειρά Ασκήσεων στο μάθημα «Οργάνωση Υπολογιστών»

Ημερομηνία Παράδοσης: Παρασκευή 31/5/2024 23.00μ.μ.

Σύστημα Παράδοσης: courses.cs.ihu.gr

Μορφή αρχείου: AEM_FHW.pdf

Σκοπός της παρούσης σειράς ασκήσεων είναι να δώσει στον φοιτητή τη δυνατότητα να κατανοήσει αρχικά τη λειτουργία ενός CISC επεξεργαστή, του Motorola 68000, με τη υλοποίηση προγραμμάτων με το λογισμικό Easy68K. Στη συνέχεια δουλεύει με ένα RISC επεξεργαστή, τον ATMega328P του Arduino, και υλοποιεί προγράμματα σε υψηλό επίπεδο.

ΜΕΡΟΣ Ι

Άσκηση 1: (η λύση να δοθεί με χρήση Ψευδοεντολών – Directives – Data Section)

Να γραφτεί ένα πρόγραμμα που θα καταχωρεί στις θέσεις μνήμης $400400-$400404 τον ορμαθό BYTENUMS των ψηφιολέξεων μήκους byte $22, $14, $BB, $BA, $C4, στις επόμενες θέσεις μνήμης τον ορμαθό WORDNUMS των ψηφιολέξεων μήκους word $3A65, $5454, θα αφήνει μια μακριά λέξη κενή για την αποθήκευση του αποτελέσματος RESULT και θα καταχωρεί στις επόμενες θέσεις μνήμης την ψηφιολέξη LONGNUMS μήκους Long Word $AA55BEEF.
Τι παρατηρείται στην αποθήκευση των δεδομένων;
Υπάρχει κάποιο πρόβλημα; Αν ναι, που οφείλεται;

Άσκηση 2:

Με βάση τον Πίνακα Μνήμης 2.1 και την ακολουθία εντολών που δίνετε και αν υποτεθεί ότι πριν την εκτέλεση των εντολών τα N=0, C=0, Z=0:

- να αναγνωριστούν οι μέθοδοι διευθυνσιοδότησης των εντολών,
- να γραφεί πρόγραμμα που θα εκτελεί τις εντολές με τα δεδομένα του πίνακα (το πρόγραμμα θα πρέπει να έχει data και code section),
- να δοθεί η αρχική και τελική κατάσταση των καταχωρητών και της μνήμης στο Easy68K.

Πίνακας Μνήμης 2.1

400400	400401	400402	400403	400404	400405	400406	400407
02	19	55	45	5A	53	00	5F
400408	400409	40040A	40040B	40040C	40040D	40040E	40040F
B0	BF	C9	D1	EC	E0	DE	CA

Εντολές		**Μέθοδοι Διευθυνσιοδότησης**	
		Τελεστέου Προέλευσης	**Τελεστέου Προορισμού**
MOVE.B	**$400401,D3**	Απόλυτη Μακριών Δεδομένων	Άμεση Καταχωρητή Δεδομένων
MOVE.W	**$400400,D4**	_____	_____
MOVE.L	**$400408,D5**	_____	_____
MOVE.B	**D4,D3**	_____	_____
MOVEA.L	**$400400,A0**	_____	_____
MOVE.B	**(A0),D5**	_____	_____
MOVE.L	**(A0)+,D1**	_____	_____
MOVE.L	**#$12345678,D1**	_____	_____
MOVE.W	**-(A0),D1**	_____	_____
MOVEQ	**#9,D4**	_____	_____
MOVEQ	**#$AE,D1**	_____	_____

ΜΕΡΟΣ ΙΙ

Άσκηση 3:

Παρακάτω φαίνεται ένα πρόγραμμα που προσθέτει δύο αριθμούς **NUM1: $F4** και **NUM2: $84** που είναι αποθηκευμένοι στις θέσεις μνήμης **$400400** και **$400401** αντίστοιχα και αποθηκεύει το αποτέλεσμα στη θέση **SUM $400402**.

	Εντολές	
	ORG	$400400
NUM1	DC.B	$F4
NUM2	DC.B	$84
SUM	DS.B	1
	ORG	$400410
ADNUMS	MOVE.B	NUM1,D0
	ADD.B	NUM2,D0
	MOVE.B	D0,SUM
	END	$400410

CCR bits	Regs/Mem
X=0, N=1, Z=0, V=0, C=0	D0=$HHHHHHF0
X=1, N=0, Z=0, V=1, C=1	D0=$HHHHHH78
X=1, N=0, Z=0, V=0, C=0	SUM=$HHHHHH78

Ερώτηση:
Είναι σωστό το αποτέλεσμα; Αν δεν είναι σωστό, τροποποιήστε το πρόγραμμα ώστε να πάρετε το σωστό αποτέλεσμα.

Σημείωση:
Το αποτέλεσμα του προγράμματος που δίνετε να εξηγηθεί με τη βοήθεια και των δεικτών του καταχωρητή κατάστασης όταν:
α. Οι δύο αριθμοί NUM1 και NUM2 ερμηνεύονται ως μη προσημασμένοι αριθμοί και
β. Οι δύο αριθμοί NUM1 και NUM2 ερμηνεύονται ως προσημασμένοι αριθμοί.

Να τρέξετε το διορθωμένο πρόγραμμα και να κάνετε επαλήθευση των αποτελεσμάτων.
Να δοθεί η αρχική και τελική κατάσταση των καταχωρητών και της μνήμης στο Easy68K.

Άσκηση 4:

Ρωτήθηκε το **ChatGPT** να υλοποιήσει κώδικα που θα βρίσκει το μέσο όρο τριών αριθμών μεγέθους Word. Απάντησε τον παρακάτω κώδικα:

```
1. ORG    $1000
2. MOVE.W    #10, D0
3. ADD.W     #20, D0
4. ADD.W     #30, D0
5. MOVE.W    #3, D1
6. DIVU.W    D1, D0
7. SIMHALT
8. END    $1000
```

Εξηγήστε γραμμή προς γραμμή τον κώδικα, ποιο είναι το τελικό αποτέλεσμα ΚΑΙ τροποποιήστε τον ώστε να αποθηκεύεται αυτό στη θέση μνήμης $400400.

ΜΕΡΟΣ ΙΙΙ

Άσκηση 5:

Να γραφτεί μια υπορουτίνα που θα μετατρέπει τα περιεχόμενα των θέσεων μνήμης **$400410 - $40041F** σε δεκαεξαδικά ψηφία (το περισσότερο σημαντικό nibble είναι μηδέν). Σε κάθε θέση είναι αποθηκευμένος ένας χαρακτήρες **ASCII**.

Τα δεκαεξαδικά ψηφία να αποθηκευτούν στις θέσεις μνήμης **$400420 - $40042F** αντίστοιχα.

Θεωρήστε ότι τα περιεχόμενα θέσεων μνήμης **$400410 - $40041F** είναι οι χαρακτήρες **ASCII**:

'C', 'O', 'M', 'P', 'U', 'T', 'E', 'R', 'S', 'C', 'I', 'E', 'N', 'C', 'E', '8'

Να γράψετε το πρόγραμμα και να επαληθεύσετε τα αποτελέσματα με το Easy68K.

Να δοθεί η αρχική και τελική κατάσταση των καταχωρητών και της μνήμης στο Easy68K.

Άσκηση 6:

Στις θέσεις μνήμης **$400400-$400404** και **$400405-$400409** είναι αποθηκευμένοι αριθμοί **NUM1: $FF,$99,$29,$5A,$3B** και **NUM2: $11,$D4,$55,$EB,$99** που αποτελούνται από πέντε (5) byte ο καθένας.

Να γραφτεί ένα πρόγραμμα που θα υπολογίζει το άθροισμα τους, προσθέτοντας τα επιμέρους byte, και θα αποθηκεύει το αποτέλεσμα στις θέσεις του αριθμού NUM1.

Σημείωση:
Για τον υπολογισμό των αθροισμάτων να γίνει χρήση της εντολής ADDX.

Να τρέξετε το πρόγραμμα και να επαληθεύσετε τα αποτελέσματα.

Να δοθεί η αρχική και τελική κατάσταση των καταχωρητών, και της μνήμης στο Easy68K.

ΜΕΡΟΣ IV

Άσκηση 7:

Να γραφτεί ένα πρόγραμμα που θα διαβάζει μια συμβολοσειρά **ASCII** χαρακτήρων και θα την αντιστρέφει με χρήση της **Hardware Stack του 68000 (A7-SP)**.

Η συμβολοσειρά θα τερματίζει με το σύμβολο ASCII **CR**.

Για παράδειγμα μπορεί να χρησιμοποιηθεί η συμβολοσειρά: **'A','N','A','T','O',$0D**

Στη συνέχεια να δοθεί λύση στο ίδιο πρόβλημα με τον καταχωρητή διευθύνσεων **A3**.

Να συγκριθούν οι δύο λύσεις και να αναφέρεται διαφορές.

Να γράψετε το πρόγραμμα και να επαληθεύσετε τα αποτελέσματα με το Easy68K.

Να δοθεί η αρχική και τελική κατάσταση των καταχωρητών, της μνήμης και της στοίβας στο Easy68K.

Άσκηση 8:

Σας δίνετε ο παρακάτω κώδικας assembly M68000 και σας ζητείται να τον περιγράψετε αναλυτικά γραμμή γραμμή.

```
1.INITSP EQU    $07FEE
2.       ORG    $400400
3.START  MOVE.L #$0010,D2
4.       MOVE.L #$0011,D1
5.       MOVEA.L #$400400,A0
6.       MOVEA.L #$400406,A1
7.       MOVEA   #INITSP,SP
8.       MOVEM.L D1-D2/A0-A1,-(SP)
9.       MOVEM.L (SP)+,D4/A3/A2/D7
10.      END START
```

Να δώσετε τα τελικά δεδομένα στον σωρό.

Stack 68k				
00007FDE				
00007FE2				
00007FE6				
00007FEA				
00007FEE				

Να γράψετε το πρόγραμμα και να επαληθεύσετε τα αποτελέσματα με το Easy68K.

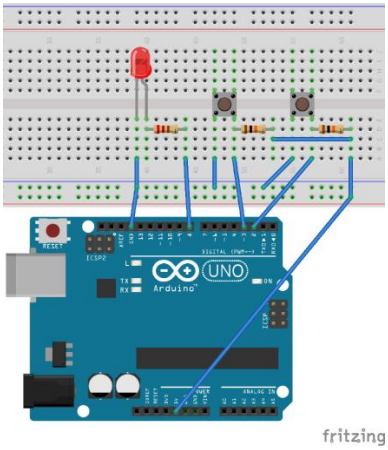
Να δοθεί η αρχική και τελική κατάσταση των καταχωρητών και της μνήμης στο Easy68K.

ΜΕΡΟΣ V

Άσκηση 9:
Να δημιουργηθεί ένα κύκλωμα που θα χρησιμοποιεί Arduino και Wiring C και θα υλοποιεί την λογική συνάρτηση F= A'B + AB + A'B'C. Για την υλοποίηση να χρησιμοποιηθούν τρεις διακόπτες ώθησης και ένα κόκκινο LED. Στο τέλος της άσκησης δίνεται βοηθητικό σχέδιο σε fritzing που μπορείτε να το επεκτείνεται. Να παραδώσετε τον κώδικα σε Arduino IDE.

Άσκηση 10:
Να δημιουργηθεί ένα κύκλωμα που θα χρησιμοποιεί Arduino και Wiring C και έναν διακόπτη (push button – με ενεργοποιημένη την Pull-up resistor) που θα ελέγχει τη λειτουργία ενός LED. Όταν θα είναι LOW θα αυξάνει τη φωτεινότητα του με τη χρήση PWM από την τιμή που βρίσκεται έως 200 (και μετά πάλι από 0 προς τα πάνω), ενώ όταν είναι HIGH θα μειώνει τη φωτεινότητα του με τη χρήση PWM από την τιμή που βρίσκεται έως 0 (και μετά πάλι από 200 προς τα κάτω).

ΒΟΗΘΗΤΙΚΟ ΣΧΕΔΙΟ:



Για την υλοποίηση σχεδίων και για την απάντηση των ασκήσεων Arduino, αποδεκτές είναι και οι προσομοιώσεις στο tinkedCAD:

https://www.tinkercad.com/