

Μετατροπές μεταξύ διαφορετικών διαμορφώσεων αριθμών κινητής υποδιαστολής

Μάθημα: Γλώσσες Περιγραφής Υλικού (CST256 / S09)

Διδάσκων: Νικόλαος Καββαδίας

`nkavn@uop.gr`

07/04/2009

Αντικείμενο της εργασίας

Αντικείμενο αυτής της εργασίας είναι η περιγραφή σε VHDL κυκλωμάτων για τη μετατροπή αριθμών κινητής υποδιαστολής από τον τύπο float (εύρους 32-bit) στον τύπο half (εύρους 16-bit) και αντίστροφα.

Ένας αριθμός κινητής υποδιαστολής χαρακτηρίζεται από τρεις ποσότητες:

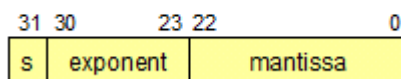
- το πρόσημο (sign ή s)
- τη βάση (mantissa ή m)
- τον εκθέτη (exponent ή e)

και δίνεται από τη γενική έκφραση

$$(-1)^{sign} \times 1.mantissa \times 2^{(exponent - bias)}$$

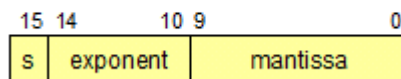
όπου bias είναι μία σταθερή πόλωσης η οποία ισούται με 127 για τον τύπο float και 15 για τον τύπο half. Οι αντίστοιχες δυαδικές κωδικοποιήσεις για τους δύο τύπους δίνονται στα Σχήματα 1 και 2.

Single floating-point format



Σχήμα 1: Δυαδική κωδικοποίηση του τύπου float

Half floating-point format



Σχήμα 2: Δυαδική κωδικοποίηση του τύπου half

Στην άσκηση δεν ενδιαφέρει η αναγνώριση ειδικών περιπτώσεων όπως των ποσοτήτων ∞ , $-\infty$ και αριθμών που δεν μπορούν να αναπαρασταθούν (NaN).

Ο κώδικας (ANSI C) του Σχήματος 3 περιγράφει τη μετατροπή half \rightarrow float και ο κώδικας του Σχήματος 4 τη μετατροπή float \rightarrow half.

```

float half_to_float(half h)
{
    unsigned int f;
    if ((h & 0x7ffff) == 0)
        return 0;

    f = ((h & 0x8000) << 16) /* sign          */
        + ((h & 0x7fff) << 13) /* exponent and mantissa */
        + ((127 - 15) << 23); /* adjust exponent      */

    return *(float*)&f;
}

```

Σχήμα 3: Κώδικας για τη μετατροπή half → float

```

half float_to_half(float f)
{
    unsigned int u = *(unsigned int*)&f;

    /* STATE 1 */
    int e = ((u >> 23) & 0xff) - 127 + 15; /* e */

    /* STATE 2 */
    int m = ((u >> 13) & 0x3ff) + ((u >> 12) & 1); /*m rounded h-up*/

    /* STATE 3 */
    if ((u & 0x1fff) == 0x1000)
        m &= 0x7fe; /* round to even */

    /* STATE 4 */
    if (m == 0x400)
    {
        m = 0;
        e++;
    } /* carry */

    /* STATE 5 */
    if (e <= 0)
    {
        return (u >> 16) & 0x8000;
    } /*underflow or zero */

    /* STATE 6 */
    if (e > 30)
    {
        return 0x7bff + ((u >> 16) & 0x8000);
    } /* overflow */

    /* STATE 7 */
    return ((u >> 16) & 0x8000) + (e << 10) + m;
}

```

Σχήμα 4: Κώδικας για τη μετατροπή float → half

Για τον κώδικα του Σχήματος 4 προτείνεται η υλοποίηση του με FSM με αριθμό καταστάσεων ίσο με 8 (οι 7 εικονιζόμενες καταστάσεις συν την αρχική ή ανενεργή κατάσταση).

Παράδοση και βαθμολόγηση της εργασίας

Στην εργασία του μαθήματος, ο φοιτητής καλείται

- να παραδώσει την περιγραφή του κυκλώματος που σχεδίασε σε VHDL
- να αναπτύξει σε κείμενο την περιγραφή της λειτουργίας του κυκλώματος
- να παρουσιάσει αποτελέσματα (π.χ. κυματομορφές, αρχεία εισόδου/εξόδου) τα οποία να αποδεικνύουν τη σωστή λειτουργία του κυκλώματος

Η εργασία παραδίδεται σε τυπωμένη μορφή (με το συνολικό κώδικα VHDL) και υποβάλλεται σε ηλεκτρονική μορφή (PDF της εργασίας + αρχεία κώδικα) στο email του διδάσκοντα. Οι φοιτητές μπορούν να παραδώσουν τις εργασίες τους το αργότερο μέχρι και την ημέρα των εξετάσεων της περιόδου Ιουνίου-Ιουλίου. Εργασία η οποία θα παραδοθεί μετά το πέρας αυτής της ημερομηνίας, δεν θα βαθμολογηθεί ώστε να ληφθεί υπόψη για τις εξετάσεις της περιόδου Ιουνίου-Ιουλίου.

Μια εργασία βαθμολογείται με άριστα το τέσσερα (4). Μη εμπρόθεσμη παράδοση εργασίας συνεπάγεται το βαθμό μηδέν (0).

Εφόσον ο φοιτητής το επιθυμεί, μπορεί να παρουσιάσει την εργασία του στην τάξη (μέχρι 12 διαφάνειες, διάρκεια παρουσίασης 15 λεπτά) την Τρίτη 16 Ιουνίου, 2009. Αυτό συνεπάγεται ότι θα πρέπει να έχει παραδώσει εγκαίρως την εργασία του, δηλαδή μέχρι και την ημέρα της παρουσίασης.

Η εργασία του μαθήματος είναι ατομική.