

АНАЛИЗ НА РЕШЕНИЕТО НА ЗАДАЧА СМУК

Анализ на задачата и стигане до решение – начинът на Искрен Чернев

Основната трудност в задачата идва от липсата на условен оператор (*if*). Тъй като трябва да трупаме сумите на отделните цветове и не знаем към кой брояч, вероятно ще трябва да трупаме във всички броячи, след подходяща модификация на теглото. Най-добрата модификация е тази, при която теглото е или 0 или себе си в зависимост от брояча. Т.е. в кода ще има нещо подобно за всяка топка:

```
cyan += weight * cyan_coef;  
magenta += weight * magenta_coef;  
yellow += weight * yellow_coef;  
black += weight * black_coef;
```

където *cyan*, *magenta*, *yellow*, *black* са променливите съдържащи сумата на теглата на съответните топки, а *_coef* променливите са внимателно сметнати да са или 0 или 1.

Ако обърнем внимание, че числата за цветовете са следните числа в шестнадесетична бройна система: 0x00ffff (*cyan*), 0xff00ff (*magenta*), 0xffff00 (*yellow*), 0x000000 (*black*). Ако извадим всички цветове от 0xffffffff, получаваме съответно 0xff0000, 0x00ff00, 0x0000ff, 0xffffffff. Сега вече лесно се вижда, че може да използваме битовете за коефициентите от по-горе.

За да вземем даден бит, е същото като деление с остатък на 2, което е просто разликата между целочисленото деление (умножено с делителя) и началното число.

И така решението се разбива на следните стъпки за всяка топка:

- намираме $base = 2^{24} - color$
- пресмятаме $yellow_coef = base \% 2$
- пресмятаме $magenta_coef = (base / 2^8) \% 2$
- пресмятаме $cyan_coef = (base / 2^{16}) \% 2$
- пресмятаме $black_coef = yellow_coef * magenta_coef * cyan_coef$
- натрупваме в съответните броячи $X += X_coef * weight$ (за *X* *yellow*, *magenta*, *cyan*, *black*)
- за изход трябва да извадим черния брояч от останалите, защото черното е броено във всички останали цветове.

Някои тънкости:

Получаване на променлива със стойност 0

```
SUB    X    X                ;;;; X = 0  
MUL    X    0                ;;;; алтернативен подход X = 0
```

Присвояване $A = B$

```
SUB    A    A                ;;;; A = 0  
ADD    A    B                ;;;; A += B
```

Намиране на $A \% B$

| | | | |
|-----|----|----|-----------------------|
| DEF | AT | | |
| SUB | AT | AT | ;;;; AT = 0 |
| ADD | AT | A | ;;;; AT = A |
| DIV | AT | B | |
| MUL | AT | B | ;;;; AT = (A / B) * B |
| SUB | A | AT | ;;;; A = A % B |

Анализ на задачата и стигане до решение – начинът на Евгений Василев

В тази задача-шега основното предизвикателство е, че микроконтролерът няма команди за цикъл, за условен преход и дори за задаване стойност на променлива. И докато липсата на средства за организиране на цикъл може да се компенсира с многократното копиране на код в програмата, то за останалите две категории липсващи команди се налага да се търсят еквивалентни по ефект комбинации от команди за целочислена аритметика.

Да обясним принципите за решаване на задачата като коментираме едно примерно решение:

Команда

```
DEF Cyan
DEF Magenta
DEF Yellow
DEF Black
DEF W
DEF C
DEF CopyC
```

```
SUB Cyan Cyan
SUB Magenta Magenta
SUB Yellow Yellow
SUB Black Black
```

```
GET W
GET C
ADD Black W
```

```
SUB CopyC CopyC
ADD CopyC C
```

```
DIV CopyC 16776960
```

```
MUL CopyC W
```

```
ADD Yellow CopyC
```

Коментар

Променливи за суматори на тегла на топките (по една за всеки от цветовете), за тегло и цвят на поредната разглеждана топка (W и C), както и копие на цвета. Всъщност при анализа на поредната топка ще се работи с копието на цвета (CopyC).

Нулираме суматорите на теглата. Ефектът може да се постигне и като умножим променливите с 0.

Първата топка

Прочитаме данни за първата топка. Добавяме теглото към „черните” – така Black ще натрупа теглата на всички топки. Накрая от Black ще извадим теглата натрупани в другите суматори и ще получим теглото само на черните.

Копираме стойността на цвета в CopyC
По-общо казано: за да зададем конкретна стойност на променлива, първо я нулираме и после към нея добавяме конкретната стойност.

Започваме анализа на цвета най-напред с най-големия код („жълто”). Резултатът от операцията е 1, ако цветът на топката е жълт, или 0 за всеки останал случай.

След операцията CopyC съдържа теглото на топката, ако тя е жълта или 0 за останалите случаи.

Добавяме към суматора за жълтите. Ако топката не е

| SUB | W | Сорус | жълта, Сорус е 0 и така суматорът не се променя! (не натрупва тегло на топка с друг цвят) Ако топката е жълта, след операцията W става 0 или не се променя при друг цвят на топката – проявява се ефектът от „преливане” на теглото в съответния суматор. | | | | | | | | | | | | | | | | |
|------------|---------|----------|--|------|-------|---|----------|------------|----|---|---|--------|---|---|---|------------|---|---|---|
| SUB | Сорус | Сорус | Анализът за пурпурна топка започва с ново копиране на кода на цвета в Сорус Резултатът от операцията е: ≥1 – ако топката има „по-голям” цвят =1 – ако топката е пурпурна =0 – ако топката има „по-малък” цвят Този резултат трябва да се разглежда съвместно с текущото състояние на W – теглото може да е „прелято” в суматор за „по-голям” цвят, което дава следните варианти за следващата операция: | | | | | | | | | | | | | | | | |
| ADD | Сорус | С | | | | | | | | | | | | | | | | | |
| DIV | Сорус | 16711935 | | | | | | | | | | | | | | | | | |
| MUL | Сорус | W | <table border="1"> <thead> <tr> <th>цвет</th> <th>Сорус</th> <th>W</th> <th>резултат</th> </tr> </thead> <tbody> <tr> <td>„по-голям”</td> <td>≥1</td> <td>0</td> <td>0</td> </tr> <tr> <td>пурпур</td> <td>1</td> <td>W</td> <td>W</td> </tr> <tr> <td>„по-малък”</td> <td>0</td> <td>W</td> <td>0</td> </tr> </tbody> </table> | цвет | Сорус | W | резултат | „по-голям” | ≥1 | 0 | 0 | пурпур | 1 | W | W | „по-малък” | 0 | W | 0 |
| цвет | Сорус | W | резултат | | | | | | | | | | | | | | | | |
| „по-голям” | ≥1 | 0 | 0 | | | | | | | | | | | | | | | | |
| пурпур | 1 | W | W | | | | | | | | | | | | | | | | |
| „по-малък” | 0 | W | 0 | | | | | | | | | | | | | | | | |
| ADD | Magenta | Сорус | Ако топката е пурпурна, „преливаме” теглото ѝ в суматора. Ако не е – в суматора постъпва 0, а теглото или вече е „прелято” в суматор на „по-голям” цвят, или остава за добавяне в суматор на „по-малък” цвят. | | | | | | | | | | | | | | | | |
| SUB | W | Сорус | | | | | | | | | | | | | | | | | |
| DIV | С | 65535 | Аналогично постъпваме и при анализа за цианова топка. Понеже е последен цвят в анализа, няма необходимост да работим с копие на цвета. Остава горния фрагмент от програмата, в който обработихме данните за първата топка, да се копира по веднъж за оставащите 7 топки и достигаме до финалния момент, когато се извеждат резултатите. | | | | | | | | | | | | | | | | |
| MUL | W | С | | | | | | | | | | | | | | | | | |
| ADD | Cyan | W | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | |
| PUT | Cyan | | | | | | | | | | | | | | | | | | |
| PUT | Magenta | | | | | | | | | | | | | | | | | | |
| PUT | Yellow | | | | | | | | | | | | | | | | | | |
| SUB | Black | Cyan | | | | | | | | | | | | | | | | | |
| SUB | Black | Magenta | | | | | | | | | | | | | | | | | |
| SUB | Black | Yellow | | | | | | | | | | | | | | | | | |
| PUT | Black | | | | | | | | | | | | | | | | | | |

Автори: Евгений Василев, Искрен Чернев