



جمع و تفریق اعداد مکمل ۲

طراحی واحد منطق و حساب

Arithmetic logic unit (ALU) design

© تمامی اطلاعات موجود در این سند متعلق به دانشگاه صنعتی امیرکبیر بوده و حقوق قانونی آن محفوظ است.



جمع و تفریق اعداد مکمل ۲



جمع و تفریق اعداد مکمل ۲

ثابت می شود که:

- همه جمع کننده های بی علامت، می توانند اعداد مکمل ۲ را نیز جمع بزنند. (اثبات کنید)
- شرط سرریز شدن فرق خواهد داشت.

- همه تفریق کننده های بی علامت، می تواند اعداد مکمل ۲ را تفریق کنند. (اثبات کنید)
- شرط سرریز شدن فرق خواهد داشت.



شرایط سرریز شدن (پاسخ غلط) در محاسبات جمع دو عدد مکمل ۲

هنگام جمع دو عدد مکمل ۲

$$A'_{n-1} \cdot B'_{n-1} \cdot S_{n-1} + A_{n-1} \cdot B_{n-1} \cdot S'_{n-1} = 1$$

○ (مثبت با مثبت بشود منفی) یا (منفی با منفی بشود مثبت)

○ (رقم نقلی به ستون آخر منتقل شود و مصرف شود) یا (رقم نقلی به ستون آخر منتقل نشود ولی تولید شود)

$$C_{n-1} \oplus C_{n-2} = 1$$

○ هر دو شرط بالا هم ارز هستند (قابل اثبات است)

مثال: مشخص کنید در فضای ۴-بیتی، کدام سرریز می‌شود و کدام سرریز نمی‌شود؟

0101	
0011	+

1011	
1101	+

0101	
0011	+

1110	
1101	+



شرایط سرریز شدن (پاسخ غلط) در محاسبات **تفریق** دو عدد **مکمل ۲**

عمل تفریق با استفاده از جمع $A + B' + 1$ بدست می‌آید، لذا این محاسبات جمع نباید سرریز شود.

مثال: مشخص کنید در فضای ۴-بیتی، کدام سرریز می‌شود و کدام سرریز نمی‌شود؟

0101
0011 -

1011
1101 -

0101
0011 -

1110
1101 -



جمع بندی سرریز شدن جمع و تفریق در بی علامت و با علامت

تفریق	جمع	سرریز شدن
Borrow = 1	Carry = 1	بی علامت
با تغییر کوچک شبه جمع	$C_{n-1} \otimes C_{n-2} = 1$ $\overline{A}_{n-1} \cdot \overline{B}_{n-1} \cdot F_{n-1} + A_{n-1} \cdot B_{n-1} \cdot \overline{F}_{n-1}$	با علامت



نکته و سوال در پردازنده‌های امروزی x86

➤ در پردازنده‌های امروزی، به دلیل یکسان بودن سخت‌افزارهای جمع و تفریق، هر دو نوع بی‌علامت و مکمل ۲ پیاده‌سازی می‌شوند.

➤ با توجه به پرچم‌های **Z, S, P, O, C** در **ALU** به سوالات زیر پاسخ دهید:

(1) پرچم سرریز **O** بر اساس منطق بی‌علامت ساخته می‌شود یا باعلامت؟ چرا؟

(2) هر یک از پرچم‌ها چگونه ساخته می‌شود؟



چرا کاربران در زبان‌های برنامه‌نویسی به سرریز شدن توجه ندارند؟

```
int a, b, c;
```

```
...
```

```
main ()
```

```
{
```

```
...
```

```
a = b + c;
```

```
if (overflow) exit(0)
```

```
....
```

```
}
```

```
a DW ?
```

```
b DW ?
```

```
c DW ?
```

```
...  
SEGMENT CODE ...
```

```
...  
ADD a, b, c ;a=b+c
```

```
JO finish
```

```
....
```

```
finish:
```

```
END SEGMENT
```

```
uint a, b, c;
```

```
...
```

```
main ()
```

```
{
```

```
...
```

```
a = b + c;
```

```
if (overflow) exit(0)
```

```
....
```

```
}
```

```
a DW ?
```

```
b DW ?
```

```
c DW ?
```

```
...  
SEGMENT CODE ...
```

```
...  
ADD a, b, c ;a=b+c
```

```
JC finish
```

```
....
```

```
finish:
```

```
END SEGMENT
```




سوال؟

