

```
void main ( ) {
```

```
    int i ;
```

```
    int [100] a ;
```

```
    int [100] b ;
```

```
    int [100] c ;
```

```
    int [100] n ;
```

a, b, c را به ترتیب

از کاربر

```
    for ( i = 0 ; i < 100 ; i++ )
```

← [i] ها را در

اندیش

```
        if ( a[i] > 2000 )
```

```
            n[i] = F ( a[i], b[i], c[i] ) ;
```

```
        else
```

```
            n[i] = G ( a[i], b[i], c[i] ) ; user = user + 1 ; }
```

```
int F ( int a , int b , int c ) {
```

```
    int z ;
```

```
    z = a * b * c ;
```

```
    return z ;
```

```
}
```

```
int G ( int a , int b , int c ) {
```

```
    int y ;
```

```
    y = a * b * c ;
```

```
    return y ; }
```

\* قابلیت های برنامه :

- حافظه

- port ورودی و خروجی

- عملیات ریاضی و منطقی

- شمارنده

- صدا کردن توابع

- کنترل شرطی اگر برقرار است توابع برنامه را ادامه بده

که اگر نیست برش شرطی داشته باشی به else  
برش + برش خسته

→ نوع متغیر int ...

→ پستابن از pointer (آدرس دهی) - استک stack

\* مجموعه دستوراتی که می پردازند + اطلاعات جانبی

→ Instruction set Architecture

\* Hardware "fix" به و در بیان تغییرات سیستم

\* for for (init-statement ; condition ; final-stat)  
for-body

≡

init - stmt ;

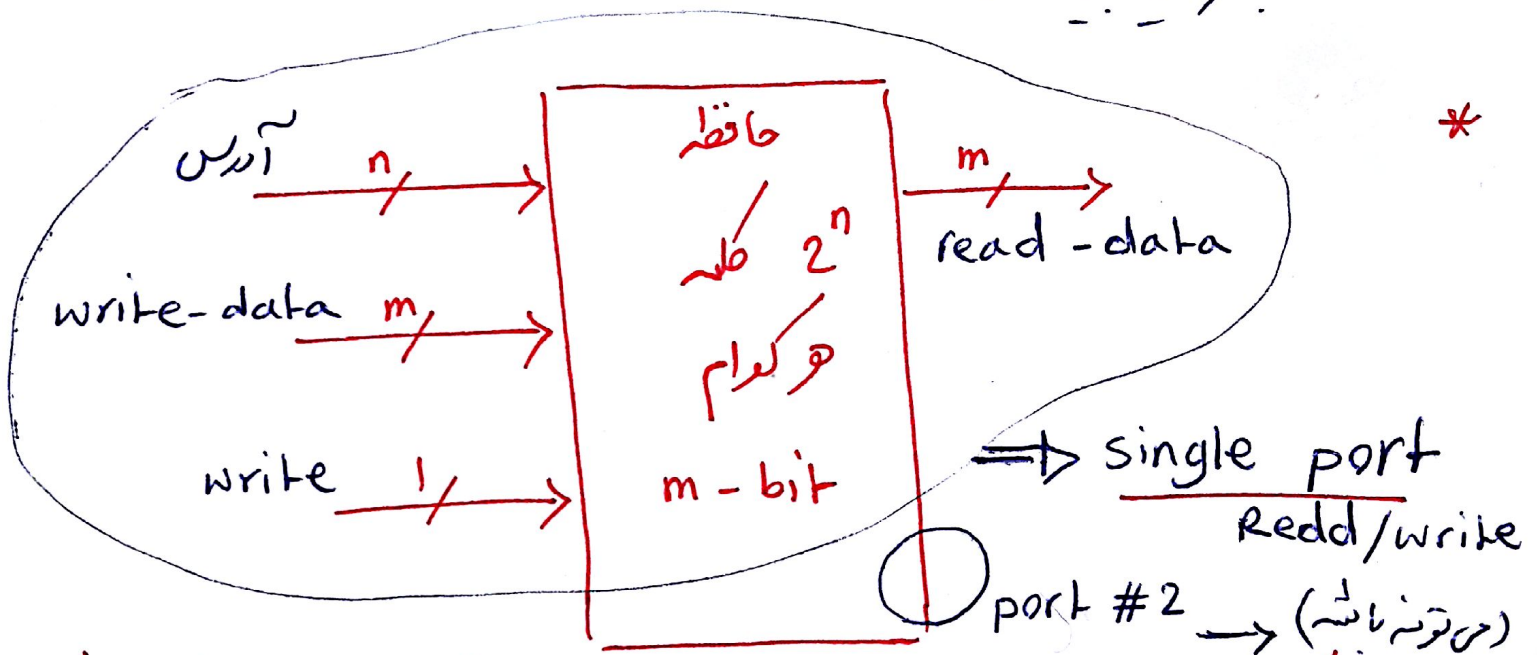
if ( condition ) {

for-body

final - stat

} goto ?

حلقه for به خودی خود قابلیت جدایی از خود دارد و کارایی آن کم است  
بنابراین سعی شود ( فقط علاوه بر پرش مشروط ، پرش جستجو  
هم دارد ) ( در برت اچ اس دی ، if ، پرش مشروط ، ... )  
بنابراین سعی شود



\* کلمه  $2^n$  ربطی به چند بیت بودن ندارد و در هر لحظه یکی از این  $2^n$  کلمه انتخاب می شود

\* بقہ port مروتق Write-only ، Read-only یا Read/write بائن .

\* read-data port مٹل حرجی یه MUX نرک ،  
که  $2^n$  ورودی داره .

\*  $n, m$  مروتق حرجی بائس ،

ولی قرار دار اینه :  $2^{30} \times 32\text{-bit}$