



# Associative Functions implemented on ClearSpeed CSX600

---

Mike Yuan



# Introduction

---

- Important for ATC applications
- Efficient implementation
- Overview
  - Setscope
  - PickOne: get, next
  - AnyResponder: any, ascAny, ascNany
  - MAX/MIN: max\_int, min\_int
- Assembler versions



# Setscope

---

- Mark the set of active PEs
- Do some operations only on enabled PEs



## Setscope (cont)

---

- Example codes:

```
//set the mask to only PE's with a 1991 model car.
```

```
  if (mycarlot.year == 1991)
  {
      mask = 1;
  }
  if (mask)
  {
      mycarlot.color = 'R';
  }
```



## Setscope (cont)

---

- Results: only PEs with years of 1991 are enabled
- Color of 1991 cars all set to 'R'



# Get

---

- Signature: mono short get (poly const char mask)
- Return the first PE number of the enabled PEs



# Get example codes

---

```
//set the mask to only PE's with a 1991 model car.  
if (mycarlot.year == 1991)  
{  
    mask = 1;  
}  
  
//get first car with the year 1991  
ONE = get(mask);  
  
//set the ONE to color M  
if (get_penum() == ONE)  
{  
    mycarlot.color = 'M';  
}
```



# Get results produced

---

- Before:

1990	L	F	1
1991	R	H	1
1992	O	T	0

- After:

1990	L	F	1
1991	M	H	1
1992	O	T	0





# Next

---

- Signature: mono short next (poly const char mask, short ONE)
- Return the PE number of the next PE in the mask



# Next example codes

---

```
//get NEXT car with 1991 year.
  ONE = next(mask,ONE);
  //set the second one to color N
  if (get_penum() == ONE)
  {
      mycarlot.color = 'N';
  }
  //skip to the fourth carwith year 1991
  ONE = next(mask,ONE);
  ONE = next(mask,ONE);

  //set the forth car with year 1991 to Z
  if (get_penum() == ONE)
  {
      mycarlot.color = 'Z';
  }
```



# Next results produced

---

- Before:

1991	G	D	1
------	---	---	---

1991	L	H	1
------	---	---	---

1991	Y	D	1
------	---	---	---

- After:

1991	N	D	1
------	---	---	---

1991	L	H	1
------	---	---	---

1991	Z	D	1
------	---	---	---



## Any

---

- Signature: mono char any (poly int *condition*);
- Test the condition *condition* on all of the enabled PEs and returns true if any of the enabled PEs return true



# Any example codes

---

```
//set mask
if (mycarlot.year == 1991)
{
    mask = 1;
}

//turn off PE's not in mask
if (mask)
{
    //if there are any red and 1991 cars
    if(any(mycarlot.color == 'R'))
    {
        //all cars turn to color T
        mycarlot.color = 'T';
    }
}
```



# Any results

---

- Before there is a:

1991	R	D	1
------	---	---	---

- After

1991	T	H	1
------	---	---	---

1991	T	D	1
------	---	---	---

1991	T	H	1
------	---	---	---



## ascAny

---

- Signature: mono char ascAny (poly int *condition*);
- Test the condition *condition* on all of the enabled PEs and returns true if any of the enabled PEs return true



# Example codes

---

```
//set mask
if (mycarlot.year == 1991)
{
    mask = 1;
}

//turn off PE's not in mask
if (mask)
{
    //if there are any red and 1991 cars
    if(ascAny(mycarlot.color == 'R'))
    {
        //all cars turn to color T
        mycarlot.color = 'T';
    }
}
```





## ascNany

---

- Signature: mono char ascNany (poly int *condition*);
- Test the condition *condition* on all of the enabled PEs and returns true if all of the enabled PEs return false



# ascNany (cont)

---

- Example codes:

```
if(ascNany(mycarlot.color=='NONE'))  
{  
  mycarlot.onlot = 0;  
}
```

- Results

1991	T	H	0
1991	T	D	0
1991	T	H	0



# Max for integers

---

- Signature of max: `int max_int(poly int value)`
- Return the maximum instance of a signed poly int value



# Min for integers

---

- Signature of min: `int min_int(poly int value)`
- Return the minimum instance of a signed poly int value



# Max/Min (cont)

---

- Example codes

```
poly int index = get_penum();  
int max_index, min_index;
```

```
max_index = max_int(index);  
printf ("The maximum of PE number is: %d\n", max_index);
```

```
min_index = min_int(index);  
printf ("The minimum of PE number is: %d\n", min_index);
```

- Results

The maximum of PE number is: 95

The minimum of PE number is: 0



# Assembler versions

---

- Use -S command line option

```
cscn -S asc.cn
```

We can obtain asc.is

- Use the `getcycles()` or `getcycles_ila()` commands to get the number of cycles