

**КОМУНИКАЦИОННА ПРОИЗВОДИТЕЛНОСТ НА ВИСОКОСКОРОСТНИ КОМУТАТОРИ ЗА СУПЕРКОМПЮТРИ**

**1. Описание на високоскоростен комутатор**

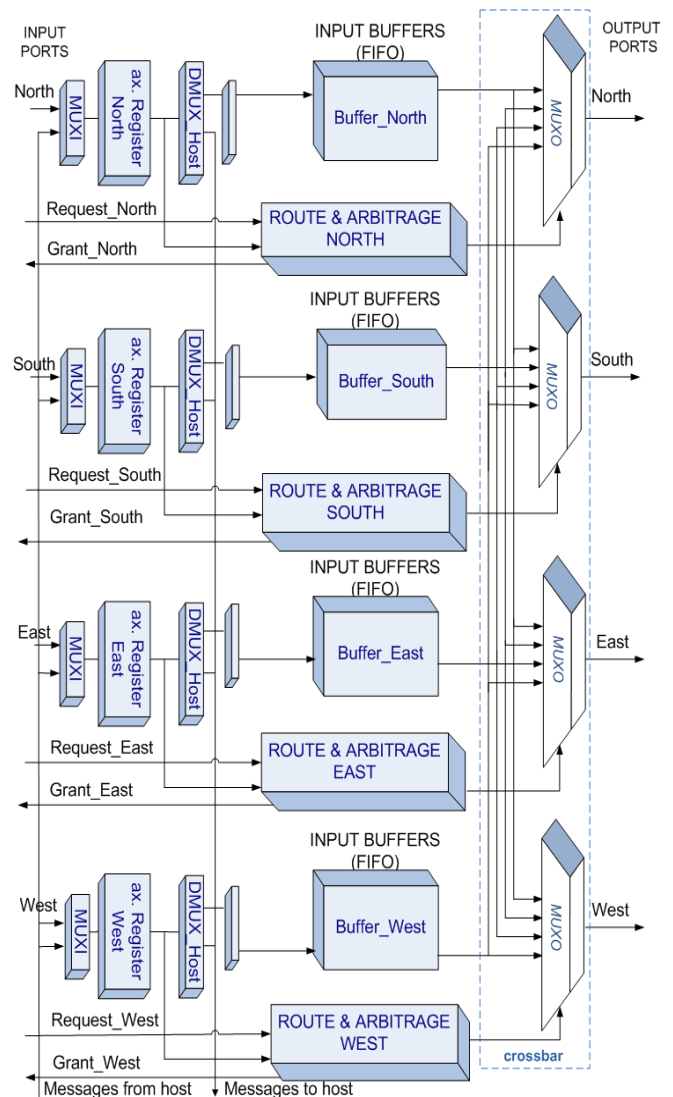
Архитектурният проект на комутатора се състои от идентични модули за комуникационен трансфер до съответните изходни портове. Всеки модул съдържа входен мултиплексор, който селектира пакет от съседен комутатор или хост.

Входният регистър съхранява 1 флит и извежда информация за рутинг алгоритъма.

Демултиплексорите (DMUX\_Host) предават напред флита до хост, ако е пристигнал сигнал на рутинг функцията за дестинацията. Ако това условие не е сила, флита се предава до съответния изходен порт според изискванията на стратегията за маршрутизация.

Капацитета на буферите е 1 флит. Буферите са свързани с изходните портове посредством „non-blocking crossbar”.

Описаниеят комутатор използва „wormhole routing” с ниска латентност.



Фиг. 1 Комутатор 4x4

## 2. Описание на симулационния модел

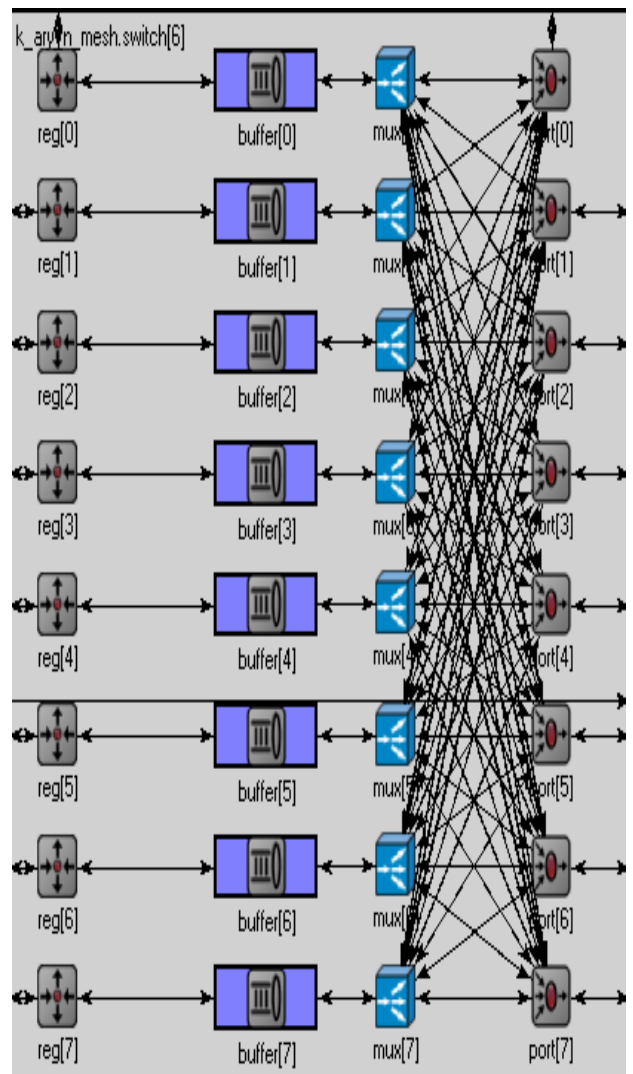
Симулационният модел е разделен на три етапа: въвеждане на регистри, FIFO буфери и неблокиращи кросбари, представени в OMNeT++ като прости модули, като целият комутатор е модул с натрупване, сложен модул в OMNeT++.

Всички прости модули, изграждащи комутатора са свързани помежду си чрез канали, както и с връзки между самите комутатори. Канали се състоят от данни за траекторията (data path) и връзки за управление (control links), които NED редактора показва само с една линия. Сложни модули: Комутаторът съдържа модул часовник Module::Clock (), виртуална функция, с която разполага всеки модул в модела. Това позволява лесно определяне на последователността на три етапа: въвеждане на регистри, буфери и изходни портове.

Всеки модул определя собствената си функционалност, но общата стратегия е: всеки модул запазва входни данни, при получаване на съобщение го препраща към

следващия модул на системата, като извиква Clock() функцията.

Хостът е прост модул в модела, генератор на трафик. Този модул поддържа своя собствен часовник, който е конфигуриран за всеки симулационен експеримент, за да ни дава различни стойности на предлаганите трафици.



Фиг. 2 Симулационен модел на комутатора

### 3. Симулация

За симулация с един комутатор се използват следните файлове:

**n\_radix\_switch.sh** - за бързо изпълнение през конзолата, като се изпълняват 10 отделни пускания (експерименти) за приложен товар (offered load) в границите от 10 до 100 %, което е зададено със следния ред в **n\_radix\_switch.ini**:

```
*.host[*].PacketRate = ({10..100 step 10})
```

Със следният ред задаваме броят пакети, които всеки хост изпраща. Този параметър влияе на точността на симулацията:

```
*.host[*].PacketsToSend = 1000
```

С този параметър на мрежата определяме радикса на комутатора (броят портове);

```
n_radix_switch.n = 8
```

ако се запише по този начин, ще се пуснат последователно 8 симулации с радикси: 4, 8, 12, 16 ... 32

```
n_radix_switch.n = ({4..32 step 4})
```

File n\_radix\_switch.ini

```
network = n_radix_switch
```

```
# this parameter specifies the switch radix and also number of receiving/sending hosts
```

```
# n_radix_switch.n = ({4..8 step 1})
n_radix_switch.n = 8
```

```
*.switch[*].*.BufferLength = 16
```

```
#
*.switch[*].*.RoutingAlgorithm = "SourceTag"
```

```
# total number of packets sent by each host
```

```
*.host[*].PacketsToSend = 1000
```

```
# packet size (in flits)
```

```
*.host[*].PacketSize = 8
```

```
#
*.host[*].TrafficPattern = "Uniform"
```

```
# specifies offered load form 0 to 100%
```

```
*.host[*].PacketRate = ({10..100 step 10})
```

```
# Flit size (in bits)
```

```
*.host[*].FlitSize = 16
```

```
*.switch[*].*.SwitchClock = 8 ns
```

```
[Config Uniform]
```

```
*.host[*].TrafficPattern = "Uniform"
```

## ВИСОКОПРОИЗВОДИТЕЛНИ КОМПЮТЪРНИ СИСТЕМИ

Брой портове за комутатор:  $p = 2k$  (или  $2k+1$ , ако към комутатора има хост)

### Работни товари / пространствени разпределения

Име	Шаблон
Равномерно разпределени дестинации	$\lambda_{sd} = 1/N$
Пермутационни	$d = \pi(s)$
побитово пермутационни	$d_i = sf(i) \oplus g(i)$
побитово комплементарни	$d_i = \neg si$
побитово реверсивни	$d_i = sb-i-1$
побитово ротация	$d_i = si+1 \bmod b$
разбъркване (Shuffle)	$d_i = si-1 \bmod b$
транспониране	$d_i = si+b/2 \bmod b$
Поцифрово пермутационни	$dx = f(sg(x))$
торнадо	$dx = sx + (\lfloor k/2 \rfloor - 1) \bmod k$
съседни	$dx = sx + 1 \bmod k$