

Subject.

Date.

NOC (نوع داده ها)

اینکه NOC به این صورت است: مثلا فرض کنید ما داریم کارانه داریم که داده ها را در آن به نوع داده ها قرار می دهیم.

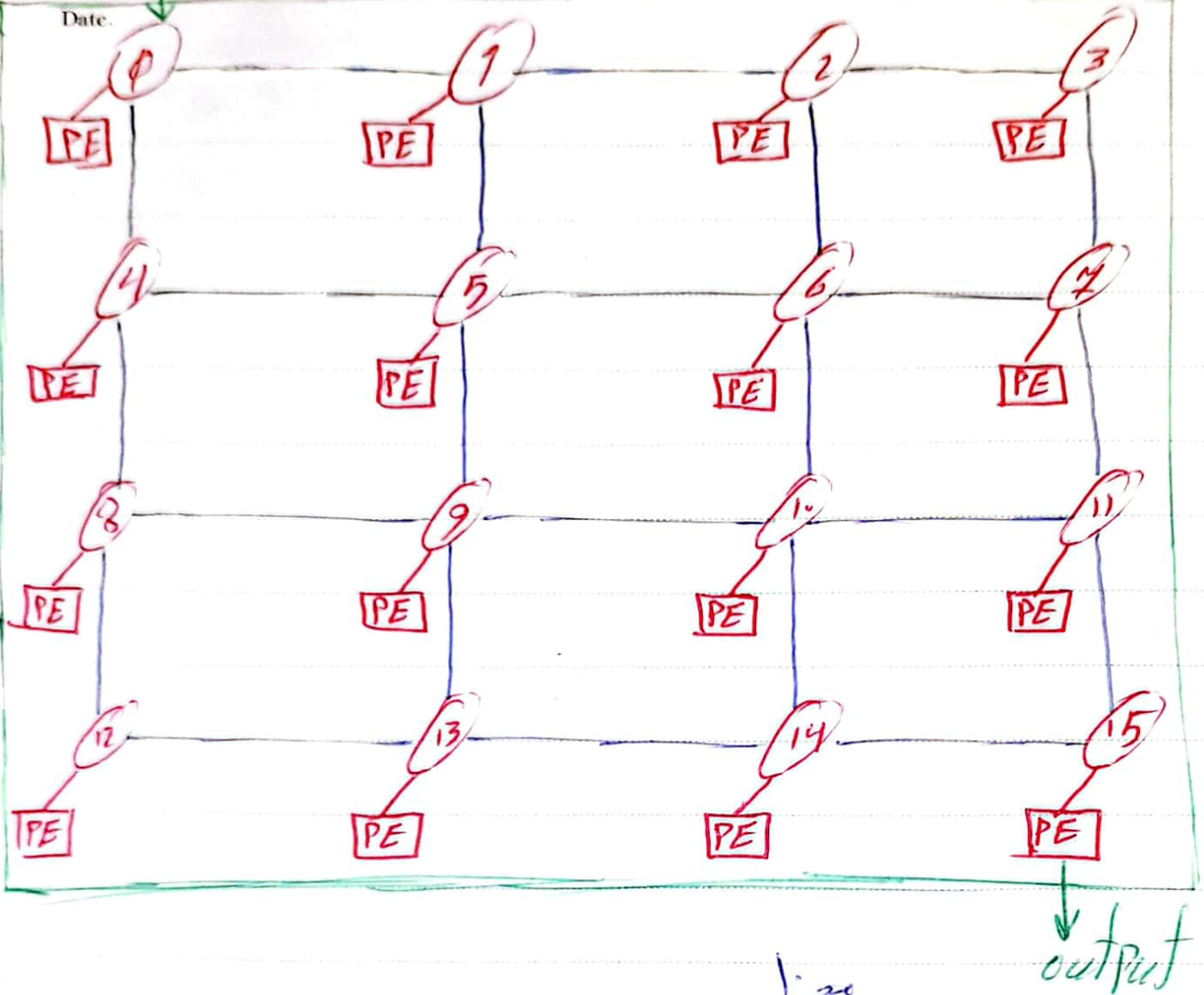
مثلا  
type I-input  
type II-input  
⋮  
type N-input

حال فرض کنید داده های  $x_1, x_2, \dots, x_n$  داریم تا این داده ها را به نوع داده ها قرار می دهیم. حال سوال اینجاست چگونه می توانیم این کارانه را مدیریت کنیم؟  
پاسخ: می توانیم از یک کارانه استفاده کنیم که به نام NOC (نوع داده ها) می باشد.

برای این مورد می توانیم از یک کارانه استفاده کنیم (یا NOC) استفاده کنیم. برای مثال کارانه را می توانیم به این صورت قرار دهیم:

Subject.  
Date.

# PROCESSOR



این یک ۱۶۰۰ unit است / <sup>جزء</sup> unit منابع محاسبه می شود

PE (Processing Element) ، Memory ، ...

این ~~یک~~ unit است که به جز <sup>جزء</sup> یک PE و <sup>جزء</sup> یک Memory است

در آن یک <sup>جزء</sup> PE وجود دارد.



۱/ حال حاضر لیست مارت های خودی در دودری خارج از ادی در خروجی نیاز است که  
 ریلر و عملیات انجام شود مثلاً

+ ادی در دودری TYPE I باید عملیات های 14, 9, 5, 4, 0

+ ادی در دودری TYPE II باید عملیات های 3, 2, 1

+ ادی در دودری TYPE III باید عملیات های

انجام شوند.

۱/ در این حالت زمانی که در دودری ریلر ~~نیاز به~~ نیاز است که ریلر و ادی اتفاق می افتد:

۱- ریلر واحد Unit عملی

۲- اگر نیاز به عملیات  $\phi$  بود، واحد PE عملی در غیر این صورت

واحد Unit عملی

۳- مست به عملیات ۲ کار عملی تا واحد Unit شود.

۴- در نهایت واحد از Unit خارج و ~~نیاز به~~ ادی خروجی قرار می گیرد.

\* حالتی که ریلر نیاز است به ریلر صورت به ریلر کار کند در نتیجه نیاز به طراحی

بر ممت برای routing یکت خارج از ریلر تا Packet ها بصورت به ریلر بین

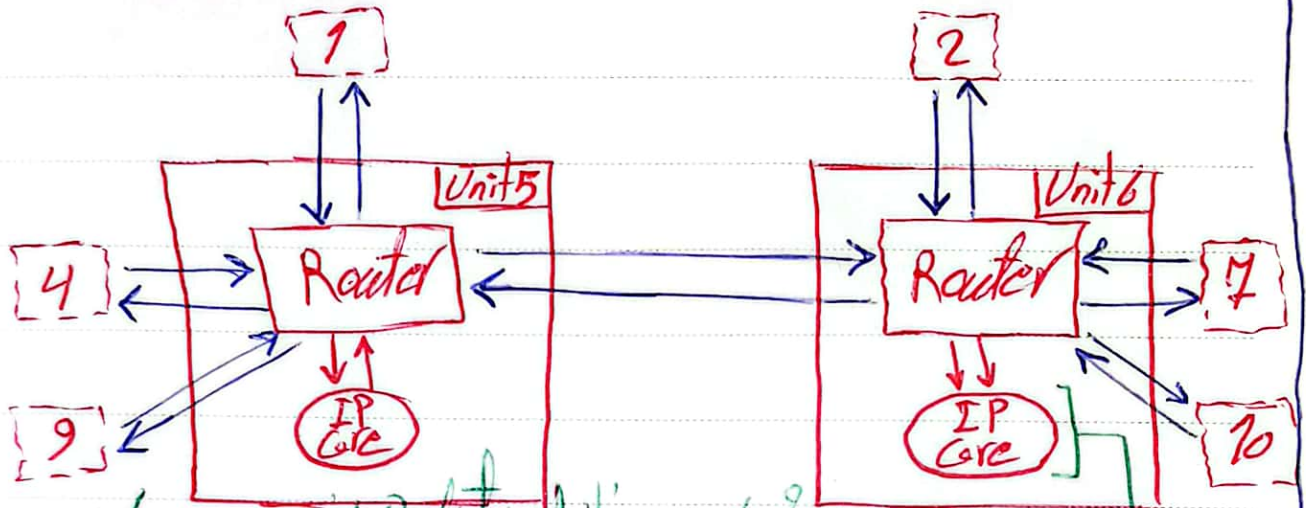
Unit حاصل شوند

\* تا اینجا باب قرار می گیرد NAC آشنا کنیم. حال نوبت به عملیات

ایش نیاز است این است که عملیات routing بین Unit ها به ریلر صورت

اتفاق می افتد.

Unit 7: packet routing  
 packet: (current unit: next unit) (dest: src)  
 در هر بار،



این سیستم برای عبور دادن packet از یک سو به سو دیگر و رسیدن آن به مقصد است.  
 Packet به صورت یونیت و آدرس در NOC قرار می گیرد (این آدرس همان مقصد است).  
 در هر بار، packet به سو Router می آید و در NOC منتقل می شود.  
 در NOC، packet به صورت Flit (Flow Control Unit) قرار می گیرد. در NOC، انواع مختلفی از Flit ها وجود دارد و به صورت زیر است:

Header Flit, Tail Flit, Payload Flit, Credit Flit, Reading Flit  
 virtual channel Flit, Error Flit

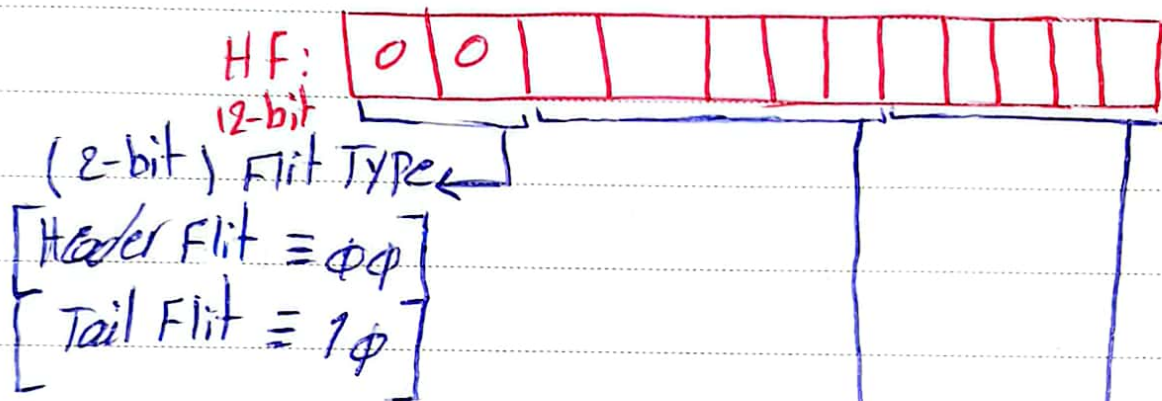


این نوع Flit ها از Flit ها به صورت زیر است:  
 Header Flit + 1  
 Tail Flit + 1



Header Flit → The Header Flit is the initial flit of a packet and contains control information such as

- + src & dest addresses,
- + routing information,
- + packet length
- + and etc.



(5-bit) src addr

(5-bit) dest addr

(5-bit) data packet

(5-bit) routing information

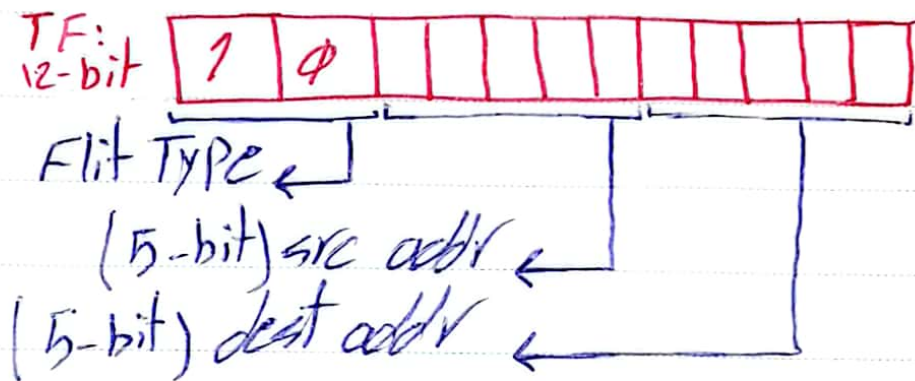
(5-bit) HF

Subject.

Date.

Tail Flit → The Tail Flit is the final flit of a packet and typically includes an end-of-packet marker or other indicators to signal the completion of packet transmission.

\* The tail flit informs the receiving router or processing element that the entire packet has been received successfully.



تا اینجا با سفتا رلی یو Unit nec و packet nec و سفتا رلی  
مالی تصدیق صورت جزئی تر سفتا رلی Router موجود سفتا  
Unit nec را مورد بررسی قرار دهیم.



Subject.

Date.

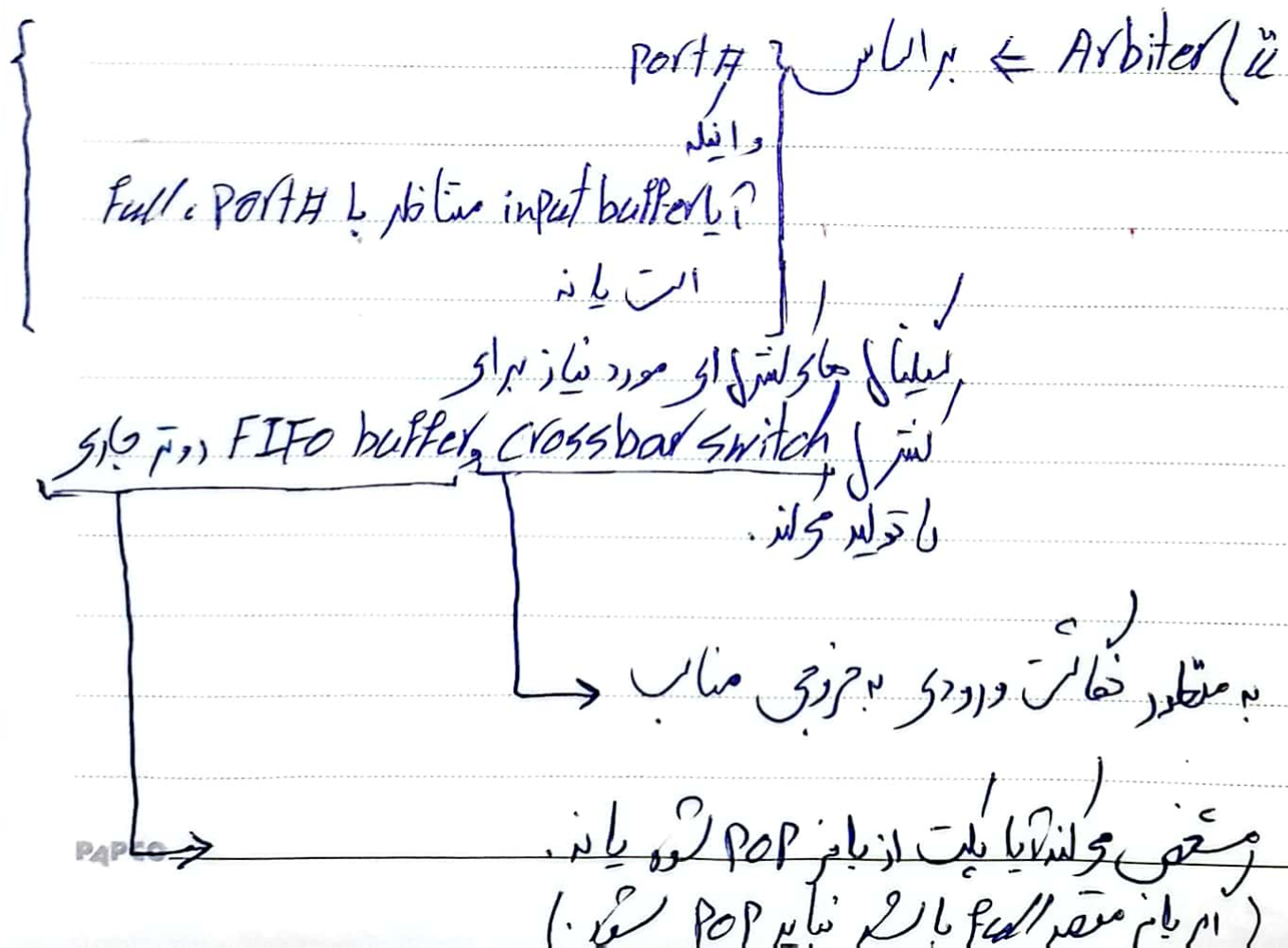
حالا نظور به می‌دانیم، از مبدأ 5 به مقصد 4 چندین مسیر وجود دارد:

$5 \rightarrow 6 \rightarrow 4$

$5 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4$

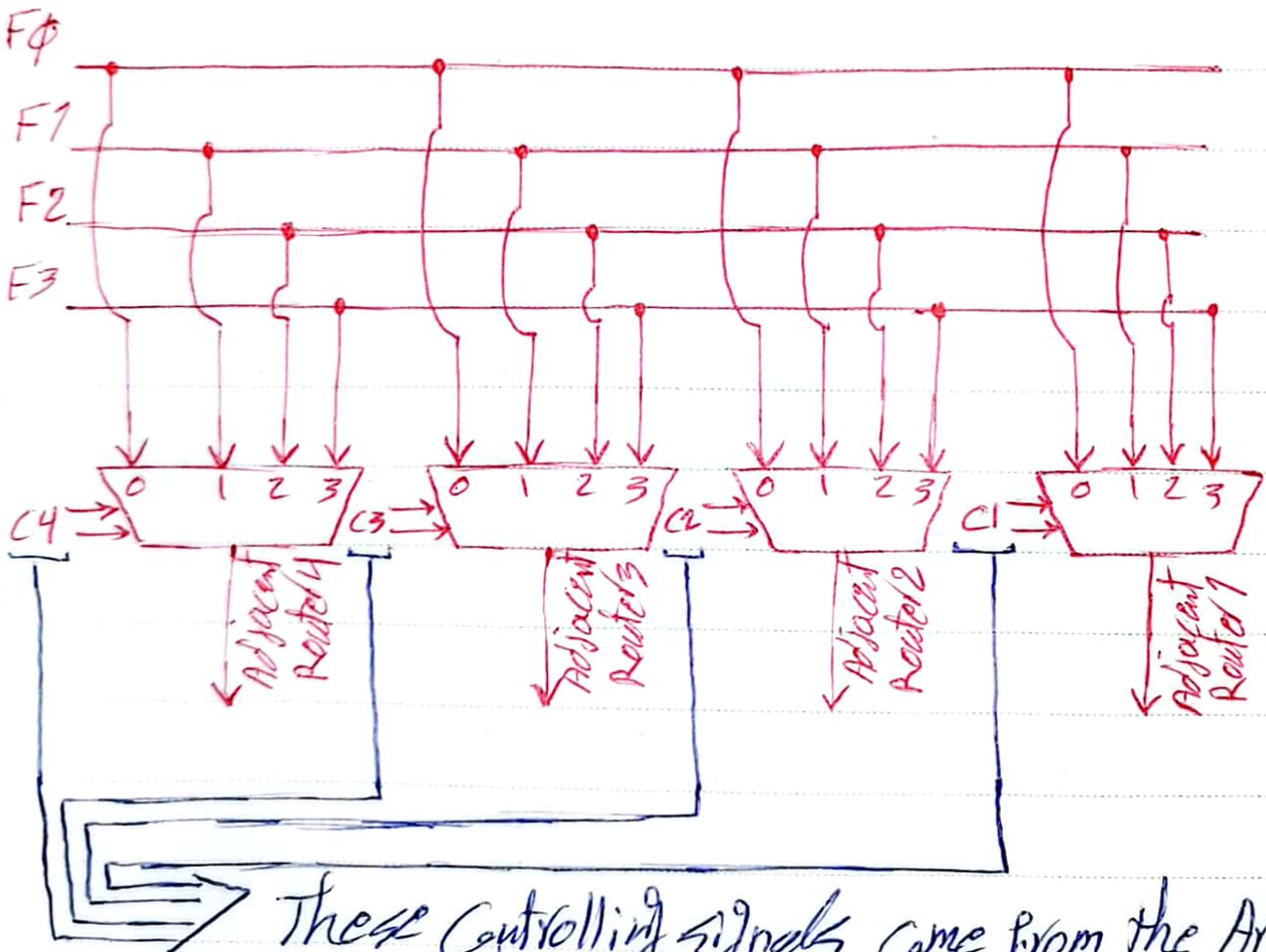
حال فرضی می‌کنیم اینجا وجود دارد این است که 5 router از کجا بفهمد که این پلست یا نه و کدام خروجی خود قرار دهد.  
برای مدیریت این مسئله به سه عامل دیگر داخل router نیاز داریم:

1) routing unit که این عامل آدرس  $src, dest$  را از Arbiter می‌گیرد و شماره پورت (port #) از پلست یا نه روی آن قرار می‌گیرد تا به Arbiter برسی می‌دهد.



Subject.  
Date.

نقشه Crossbar switch ← عملکرد این کامپیوتر به این صورت است  
 { چند چورت ورودی }  
 { چند چورت خروجی }  
 و به واسطه لینکهای خاص (شکل) ورودی و خروجی را به چورت ورودی یا به چورت خروجی مناسب متصل میکند.



These Controlling signals come from the Arbiter.

C1 ← متن میانه packet موجود در ایتر (1) صف وارد  
 Adjacent Router 1 (1) شود.

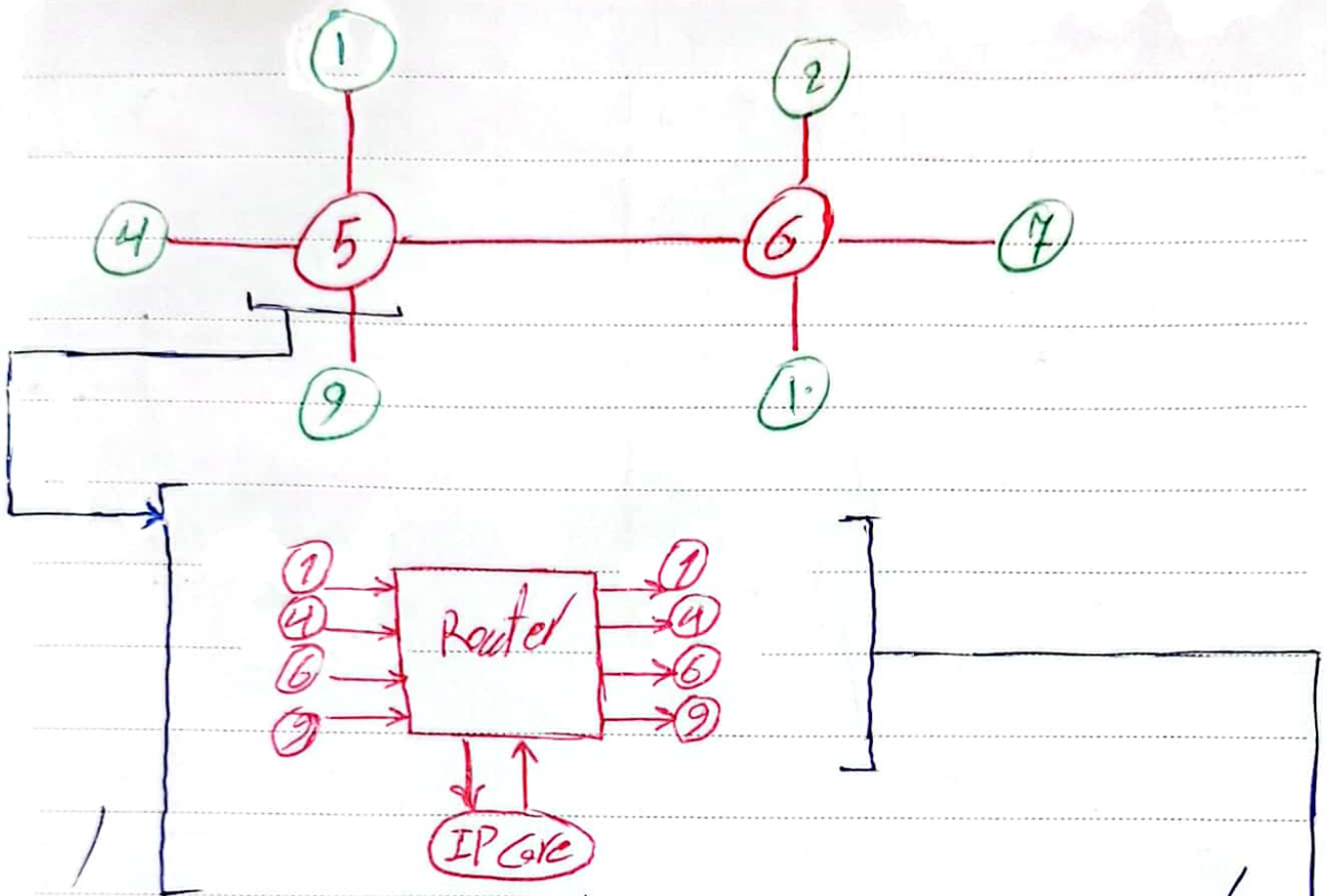
C2 ← متن میانه packet موجود در ایتر (2) صف وارد  
 Adjacent Router 2 (2) شود.



Subject:

Date:

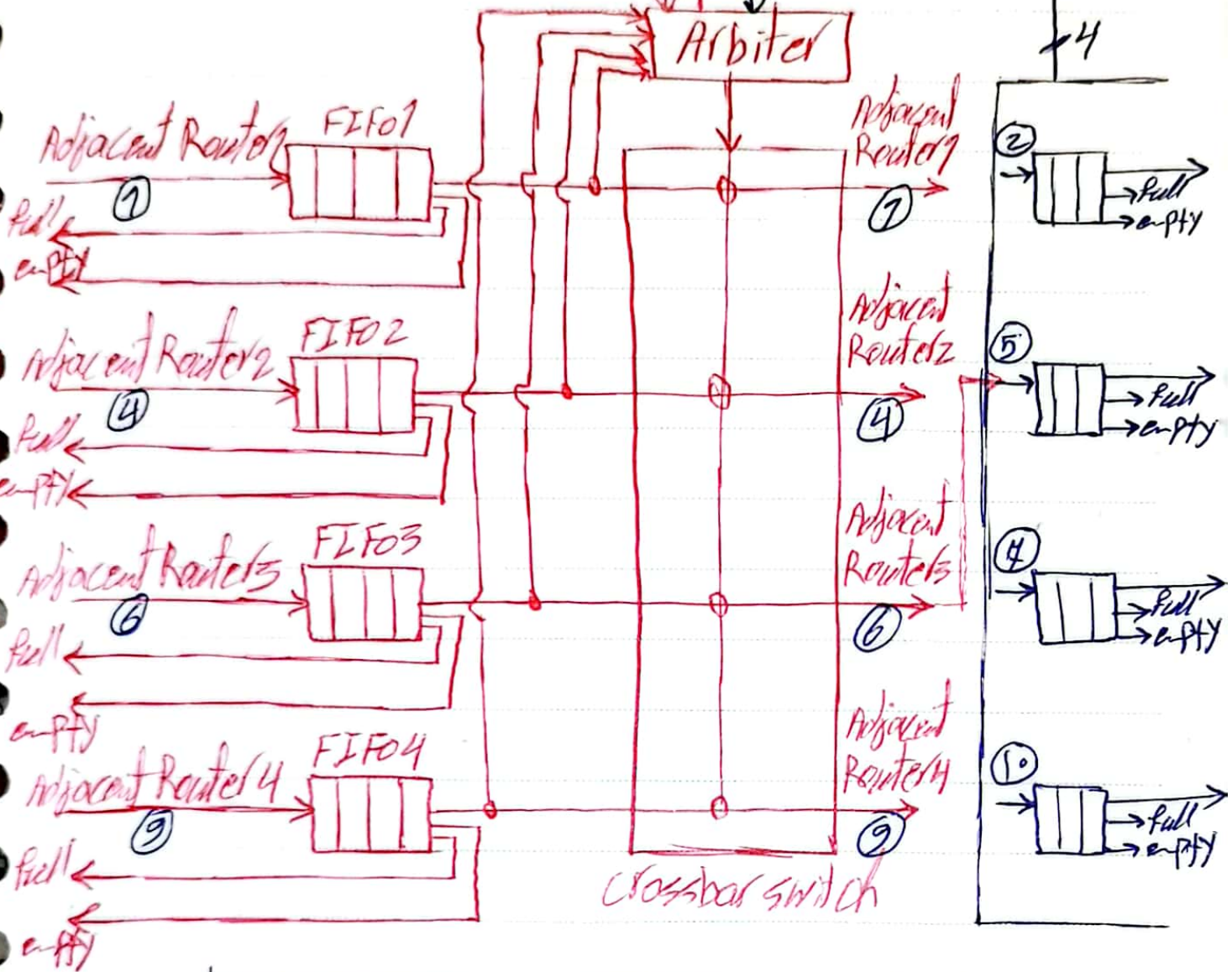
\* بطور فیزیکی، یک Router می تواند به تعدادی دستگاه به هم وصل شود.



✓  
 ← از هر یک از این دستگاه ها به یک Router می توان به هم وصل کرد و به هم وصل کرد.  
 با استفاده از یک دستگاه به هم وصل کرد.

Routing Unit

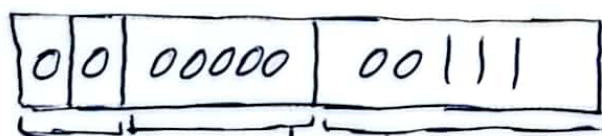
Full (2), Full (5), Full (7), Full (10)



Router 5

Router 6

طبق این ساختار، هر ورودی به router به FIFO buffer متصل است و  
 به یک صف حاوی ورودی حاضر در آن قرار می گیرند.  
 (یعنی صف حاوی ورودی از Unit مجاور)  
 حال فرض کنید Packet زیر در ابتدای FIFO1 قرار دارد.

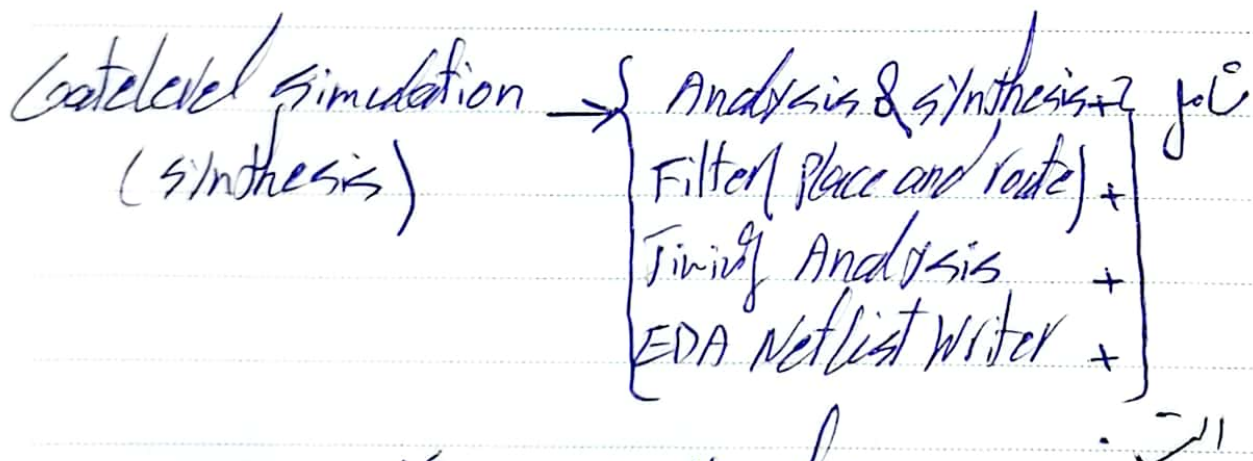
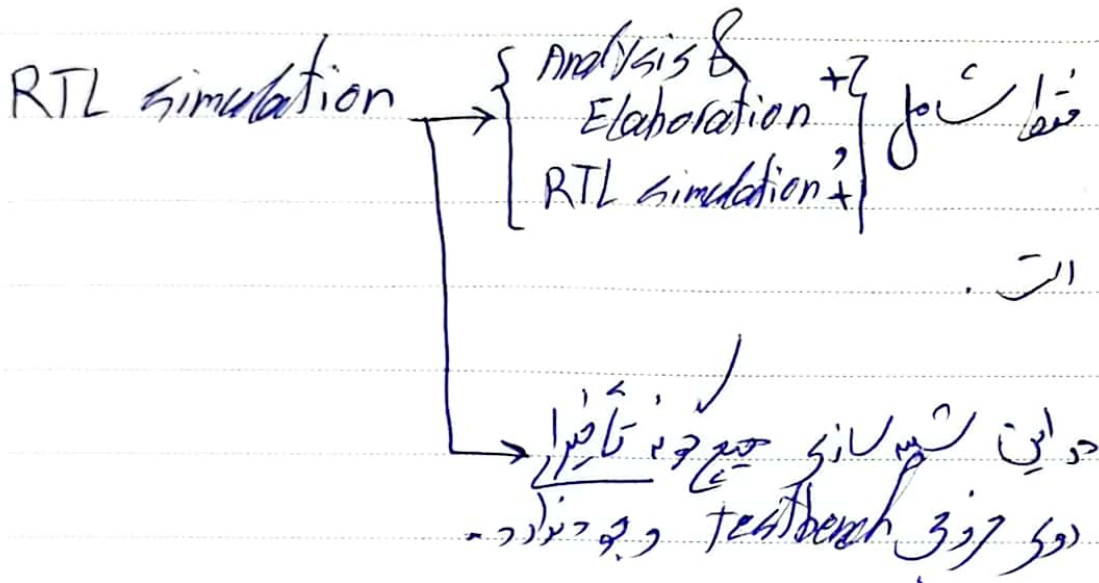


P4PCQ  
 dest = 4

از Unit 1 خارج شده  
 و الان حاضر در Unit 5 قرار دارد



⊆  $MV \times 4:1$  کامپوٹس



\* در این سیستم از testbench و تاخیر در این است (Timing model: 1.2V - 4 - 50nH)

$$t_{pdr\_out} \approx 8ms$$

$$t_{p/f\_out} \approx 9ns$$

~~Not generic - 1, 2, 4, 5~~

کامپوزیشن - register file  $\Leftarrow$  max gate level simulation تا ۱۷ns تأثیر حدوداً ۱۷ns

از طرفی ~~بسیار~~ پیچیده و حتماً عملیات ~~دو~~ <sup>۱۰۰۰</sup>،  
دتر لکه، حتماً تغییر حرفی ~~بسیار~~ <sup>glitch</sup> ~~دو~~ <sup>دو</sup>  
بیت جای مختلف به وجود می آید تا حرفی به حالت پایدار برسد.

(קולטת כבוד)

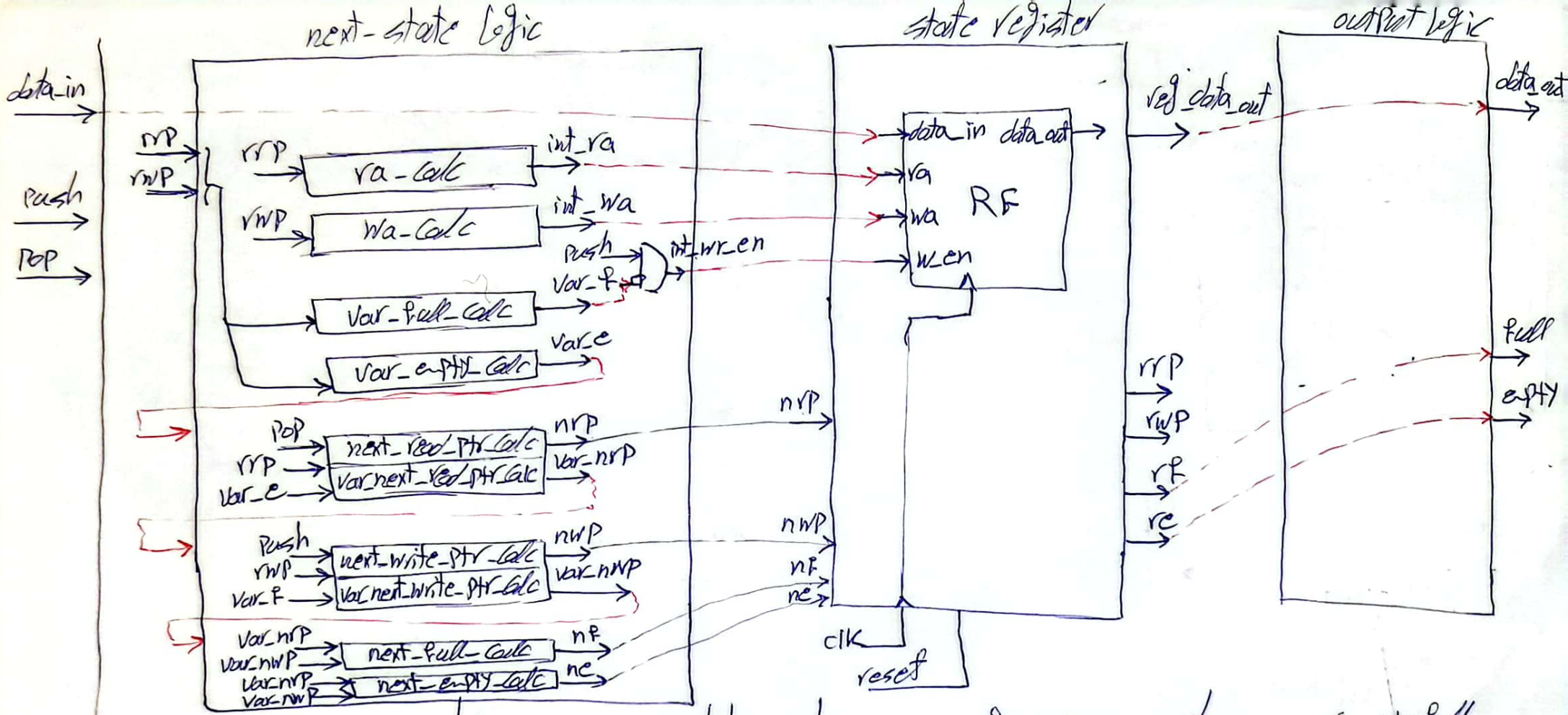
\* دقت له در نشر م. ب. ب. مقدار معمولاً در حوضی ظاهر می‌گردد.

\* «عترت» در زمان حضور ائمه از «Genetic Engineering»

$\left\{ \begin{array}{l} \text{Component} + \\ (\text{default value}) \text{ entity} + \end{array} \right\}$  is generic ~~entity~~ entity ~~set~~

با هم به هم گفتیم  
 و این EDA tool هست



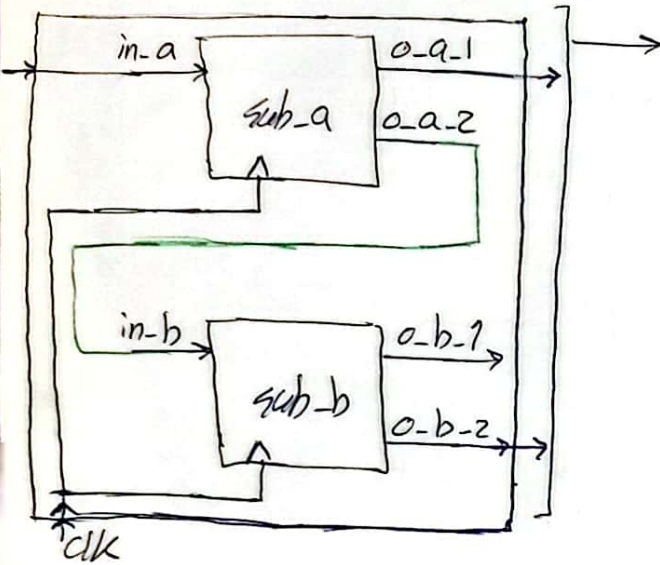


برای تعیین اینکه آیا پشته خالی است یا نه، باید وضعیت  $Var-Pull$  و  $Var-empty$  را بررسی کرد. اگر  $Var-Pull$  فعال باشد و  $Var-empty$  غیرفعال باشد، پشته خالی است. در این حالت،  $next-read-Ptr$  و  $next-write-Ptr$  باید به یکدیگر برابر شوند.

برای تعیین اینکه آیا پشته پر است یا نه، باید وضعیت  $next-Pull$  و  $next-empty$  را بررسی کرد. اگر  $next-Pull$  فعال باشد و  $next-empty$  غیرفعال باشد، پشته پر است. در این حالت،  $Var-next-read-Ptr$  و  $Var-next-write-Ptr$  باید به یکدیگر برابر شوند.

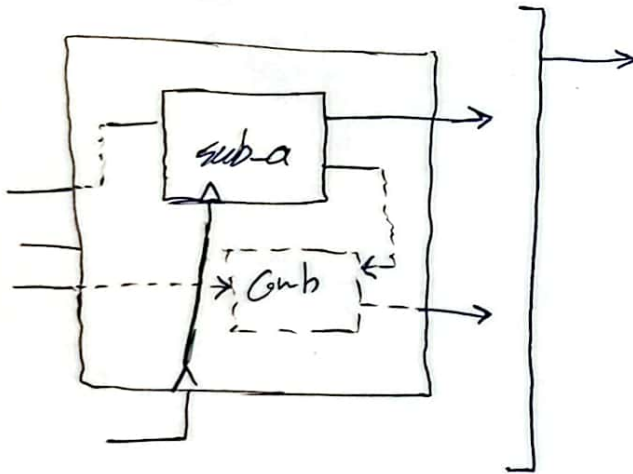
در صورتی که پشته پر باشد، باید وضعیت  $next-read-Ptr$  و  $next-write-Ptr$  را بررسی کرد. اگر  $next-read-Ptr$  و  $next-write-Ptr$  برابر باشند، پشته پر است. در این حالت،  $Var-next-read-Ptr$  و  $Var-next-write-Ptr$  باید به یکدیگر برابر شوند.

در حالتی که sub-a, sub-b هر یک زیر سیگنال متقل هستند،  
حرکت را به صورت جدا بار در سیگنال two-segment پیاده سازی  
و سپس با port mapping آنها را به هم متصل می کنند.



اما زمانی که sub-a جزئی از یک سیگنال دیگر باشد، ابتدا  
sub-a با بار در سیگنال two-segment پیاده سازی و سپس  
در باره state اصلی را به در حالی که sub-a در Component از آن است،  
بار در سیگنال two-segment پیاده سازی.

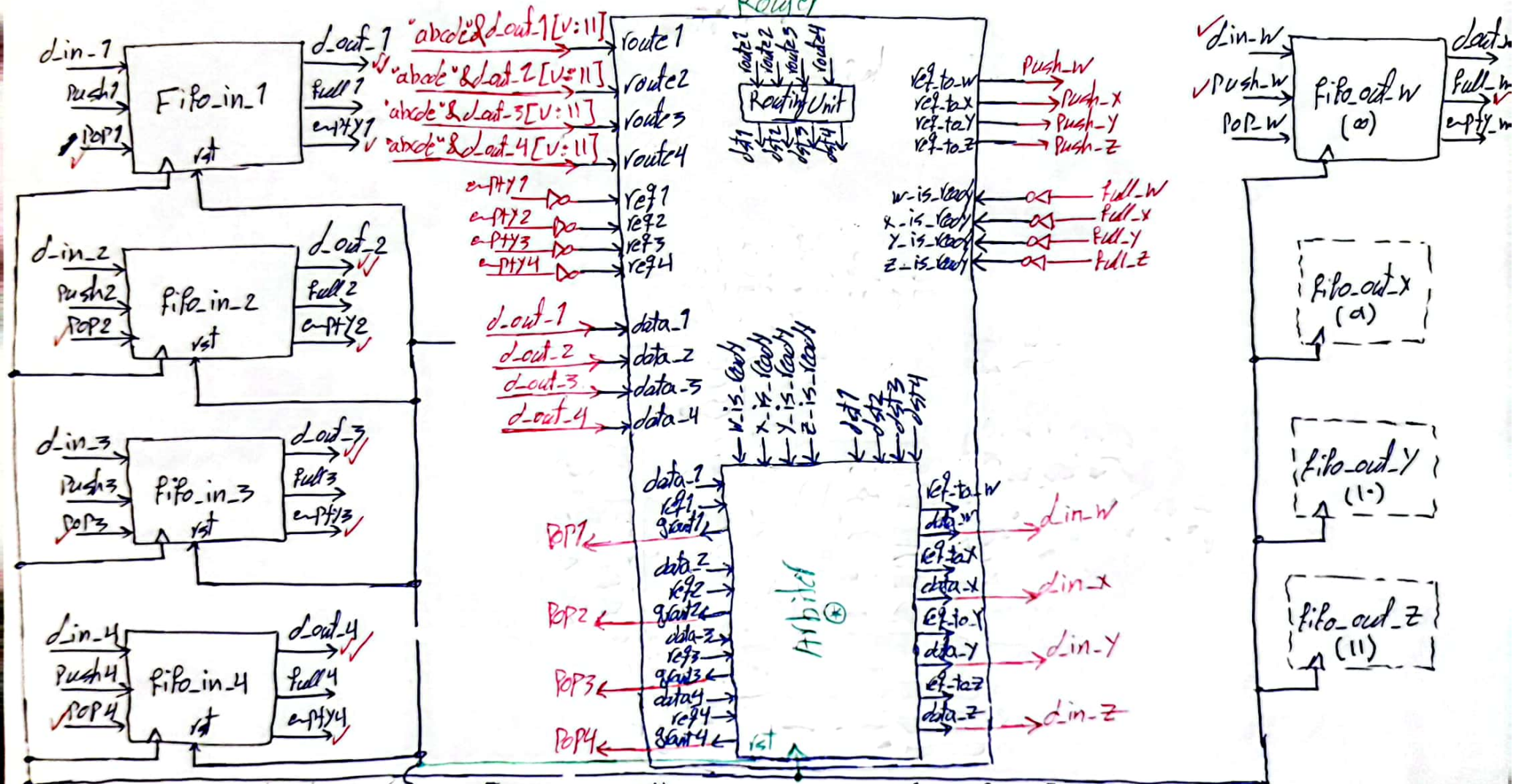
(مثل Fifo-buffer)



~~Variable assignment~~ زمانی که کاری با سیگنال داخل یک Process، بعد از Variable assignment  
از محاسبات انجام داده شده است.  
(این کار با assignment قابل انجام نیست زیرا مقدار محاسبه شده در Process به آن assignment داده می شود)



# Router

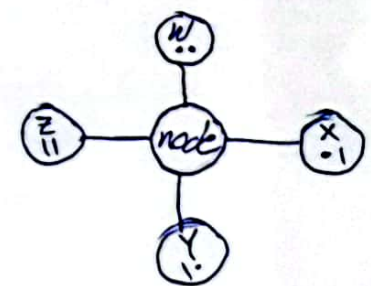


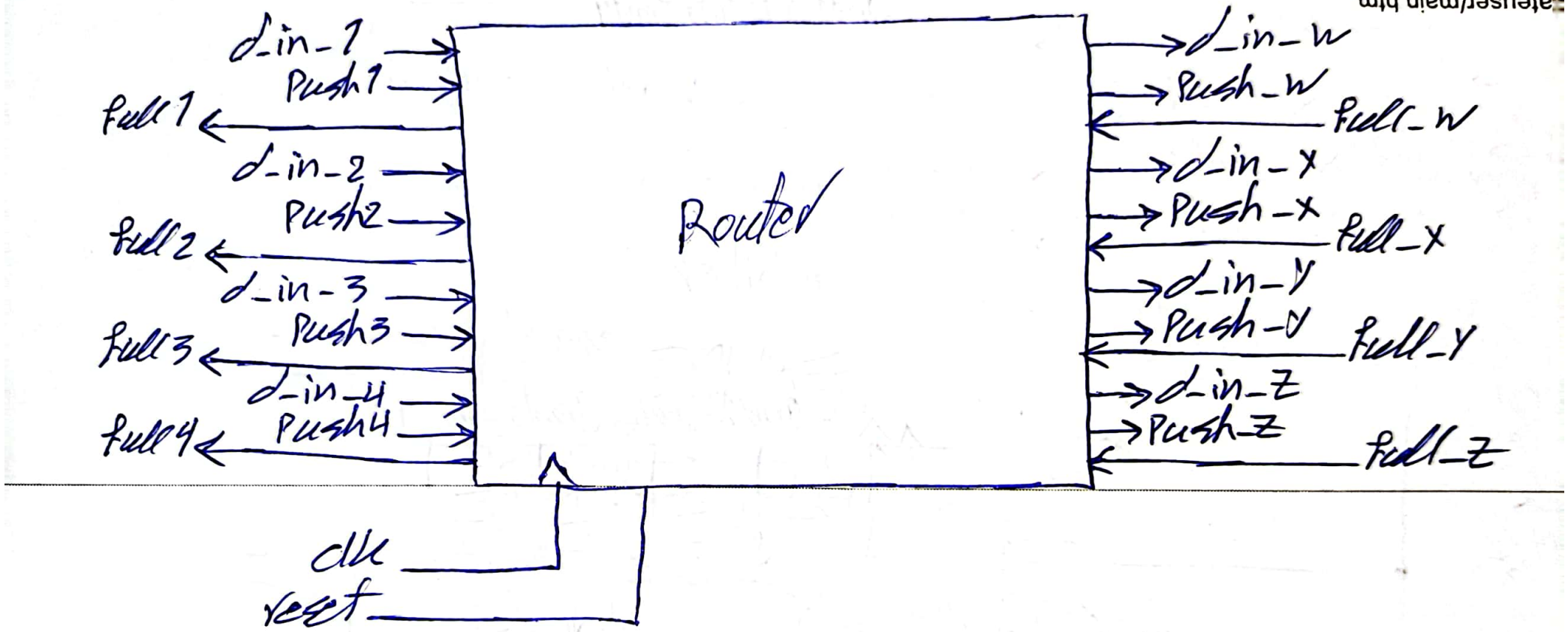
clk  
 reset  
 routing unit

```

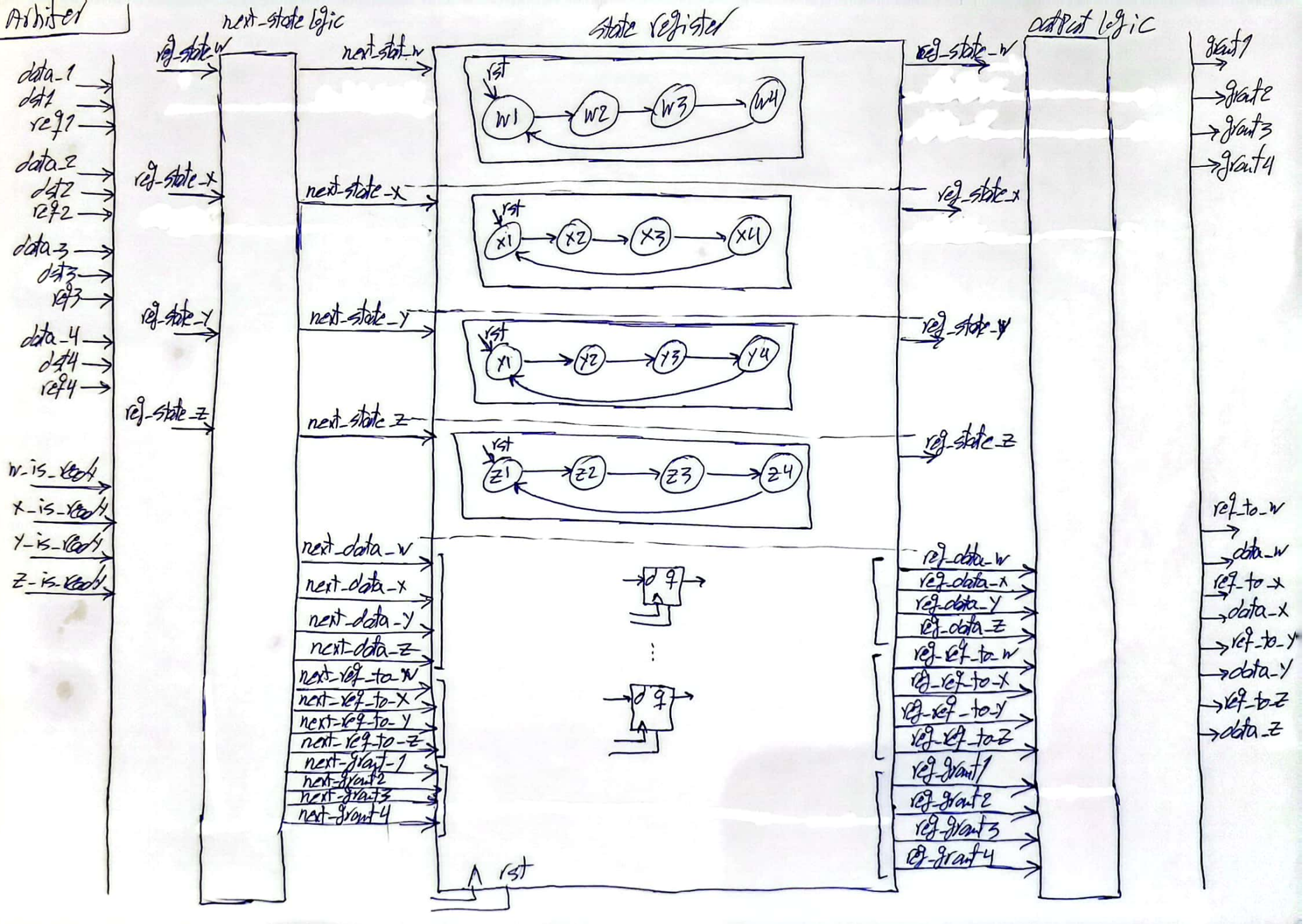
if( src = dst )
    // packet is received
    dst (src_row = dst_row)
    // move horizontally
    elif( src_col = dst_col )
        // move vertically
    else
        // do one step horizontally or vertically
        in correct direction
    
```

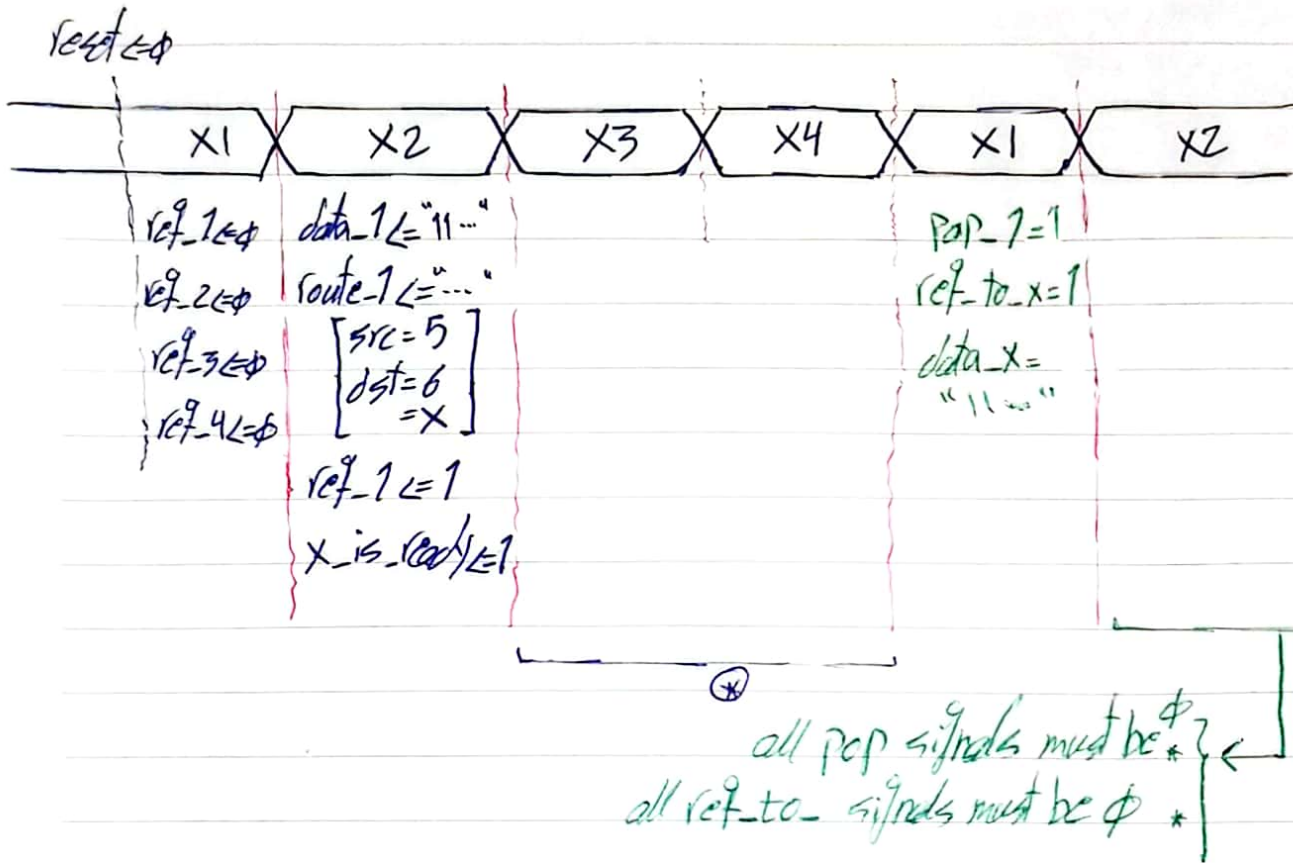
$\Phi = 0000$   
 $\tau = 00100$   
 $\Lambda = 01000$   
 $\kappa = 01100$   
 $01111 = 15$







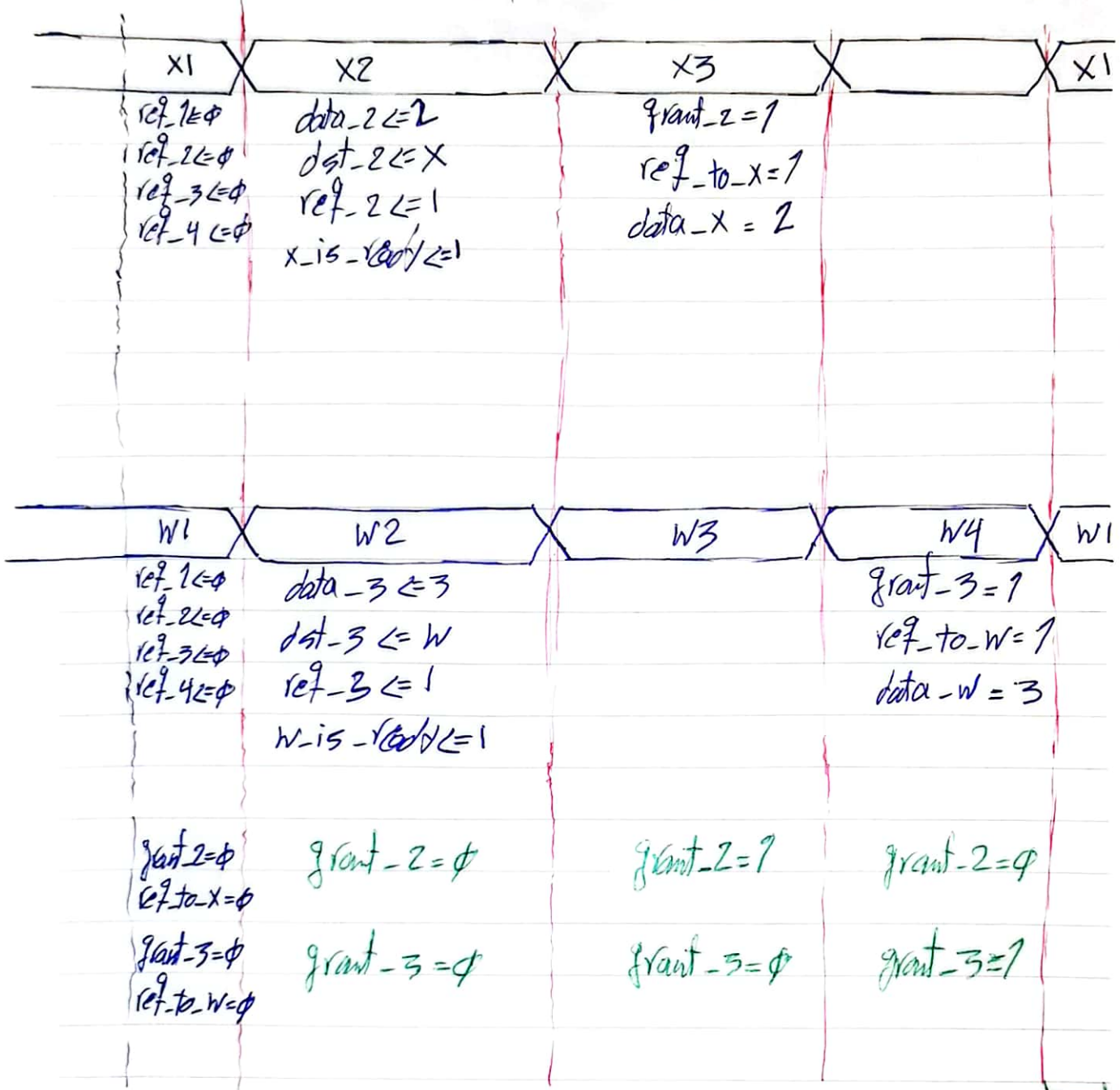




- نکته 1  $\leftarrow$  فقط جورت 1 (route-1) از routing-unit پیاده سازی شده است.
- نکته 2  $\leftarrow$  ~~دور~~ دورت دوم اولویت دهی (arbitrator) بر اساس اولویت Round Robin است.
- یعنی اگر { Fifo-1 } بفراوانده شود همان  $\phi$  خود X (خود نماد 6 با Fifo-2)
- مبدأ 5) بلت اداسی که با نرفش ایند arbitrator در state  $X^2$  با  $\phi$
- باله لاک بعدی دتای Fifo-2  $\phi$  مقصد اداسی می شود.



reset = 0



grant-2 =  $\phi$   
 ref-to-X =  $\phi$   
 grant-3 =  $\phi$   
 ref-to-W =  $\phi$

node-5

