

III КОНФЕРЕНЦИЯ FPGA РАЗРАБОТЧИКОВ

FPGA-Systems 2021.2

Доступно в записи на Youtube

Конференция в Москве



Конференция в
Санкт-Петербурге

Приходи на следующую конференцию

fpga-systems.ru/meet

Поддержи мероприятие

Способ 1

Способ 2

III конференция FPGA разработчиков

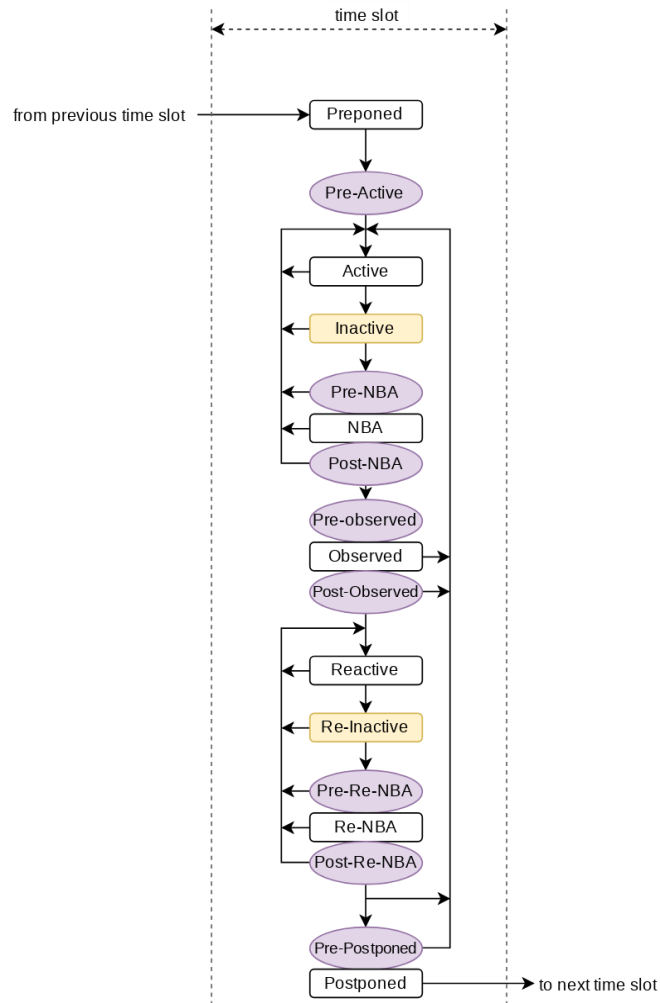
FPGA-Systems 2021.2

SystemVerilog event regions.
Как разобраться?

Данил Бычков

=

<=



PLI callbacks

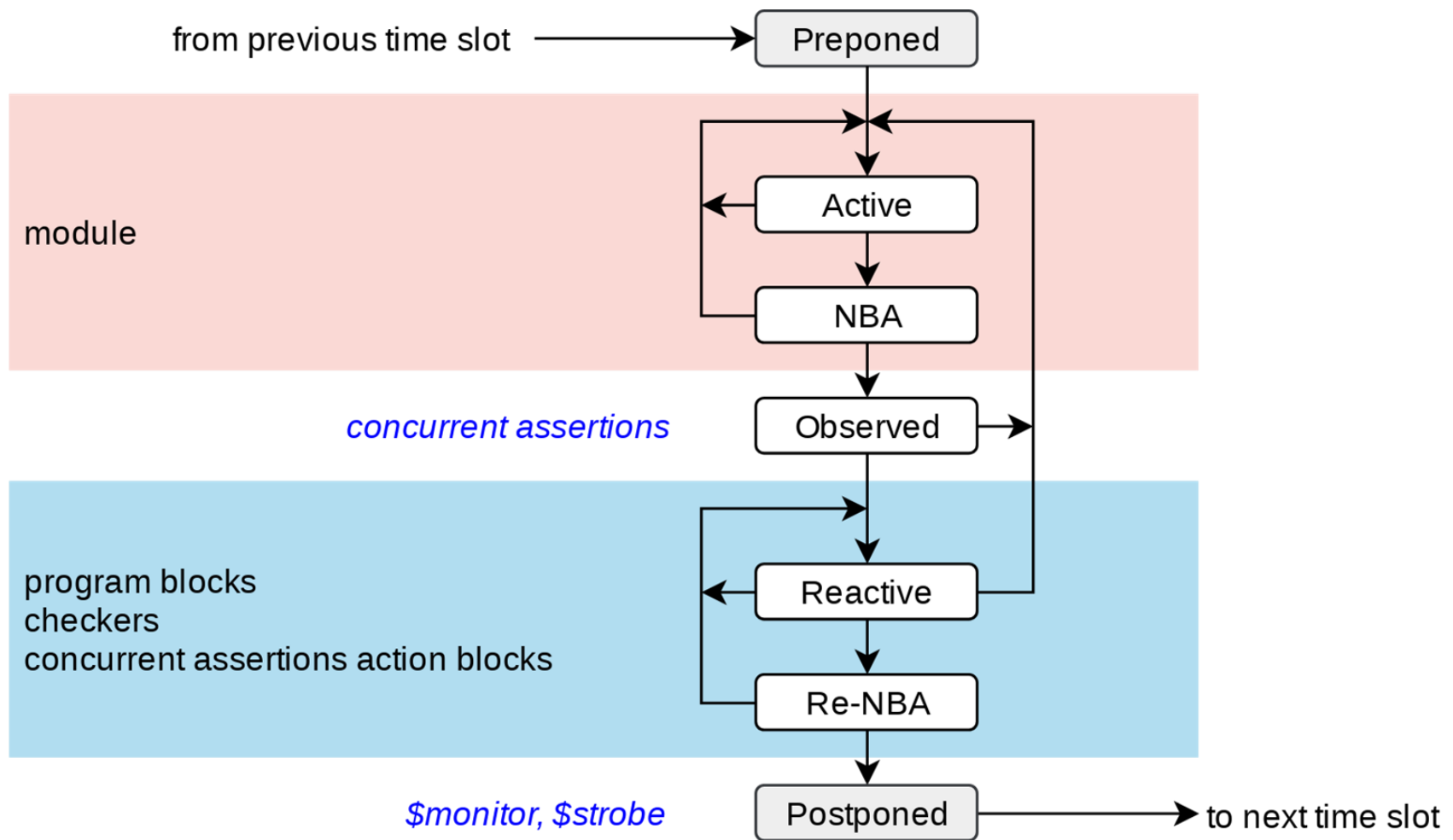
PLI region

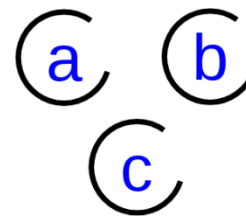
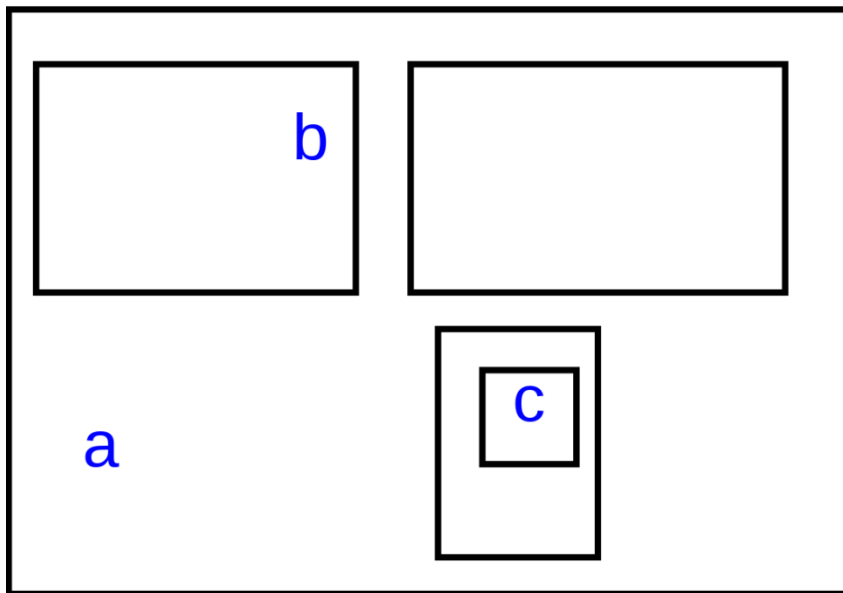
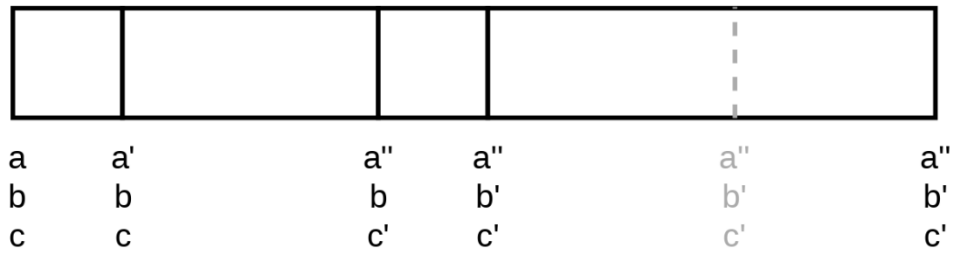
PLI callbacks, которые
назначаются на выполнения в
этих регионах
не поддерживаются
ModelSim/Questa

#0 statements

Inactive region

Хотя Inactive region
поддерживается в
ModelSim/Questa, в стандарте
SystemVerilog не рекомендуется
его использование





initial

Выполняется один раз

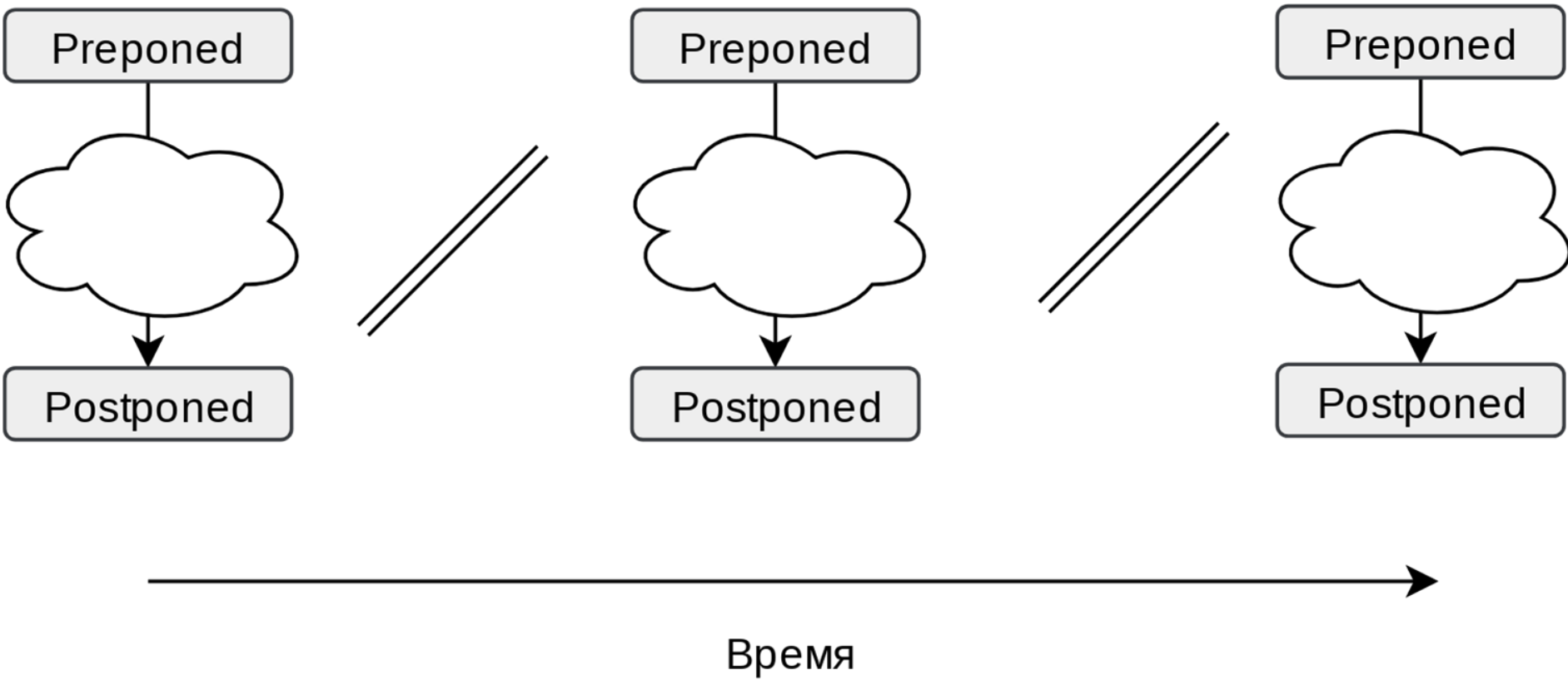
always

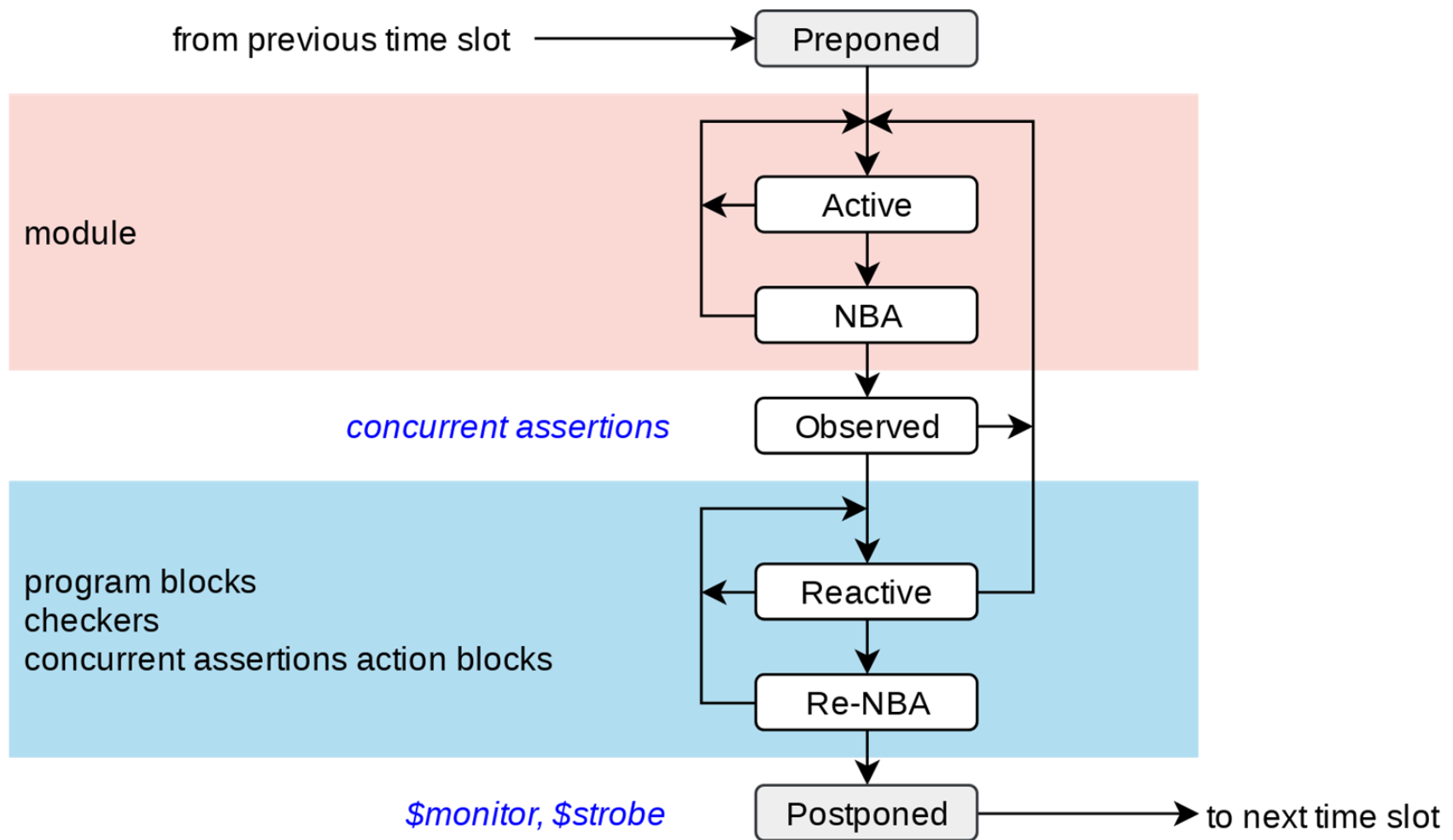
Выполняется циклически

assign

- Выполняется циклически
- Имеет особенности, когда LHS мета-типа net

основа	описание	варианты
@	ожидание по списку чувствительности	@(<u>posedge clk</u>) @ my_event @* @my_clocking_block
#	ожидание времени	#(<u>10ns</u>)
##	ожидание default clocking-блока	##1 ##0
wait	ожидание true определенной переменной	wait(my_signal)
wait fork;	ожидание окончания дочерних процессов	





Три

- Процессы, которые могут ждать
- Планировщик итераций
- Регионы внутри тайм-слота

```
integer a, b, c,  
d, e;
```

```
assign c = a;
```

```
initial  
begin  
    #20 a <= 12;  
    a <= 13;  
    #5 a = 6;  
end
```

```
initial  
begin  
    #25 e = a;  
end
```

```
always @(a)  
begin  
    b <= a + 1;  
    d = a - 1;  
end
```

```
always @(b)  
begin  
    $display(b);  
end
```

```
integer a, b, c,  
d, e;
```

```
assign c = a;
```

```
initial
```

```
begin  
    #20;  
    a <= 12;  
    a <= 13;  
    #5;  
    a = 6;  
end
```

```
always
```

```
begin  
    @(a);  
    b <= a + 1;  
    d = a - 1;  
end
```

```
initial
```

```
begin  
    #25;  
    e = a;  
end
```

```
always
```

```
begin  
    @(b);  
    $display(b);  
end
```

Не поймал событие

```
begin
  a = 25;
  #20 a = 26;
  @(a);
  b = 0; //never get here
end
```

Поймал событие

```
begin
  a = 25;
  #20 a <= 26;
  @(a);
  b = 0; // будет ждать NBA где и
          // дождется и симуляция продолжится
end
```

Не поймал named event

```
begin
  #20;
  ->evt;
  @evt;
  b = 0; //never get here
end
```

Поймал named event

```
initial
begin
  #20;
  ->>evt;
  @evt;
  b = 0; // b happily grounded
end
```

WAIT failed

```
begin
  a = 25;
  #20 a = 26;
  wait(a==26);
  b = 0;
end
```

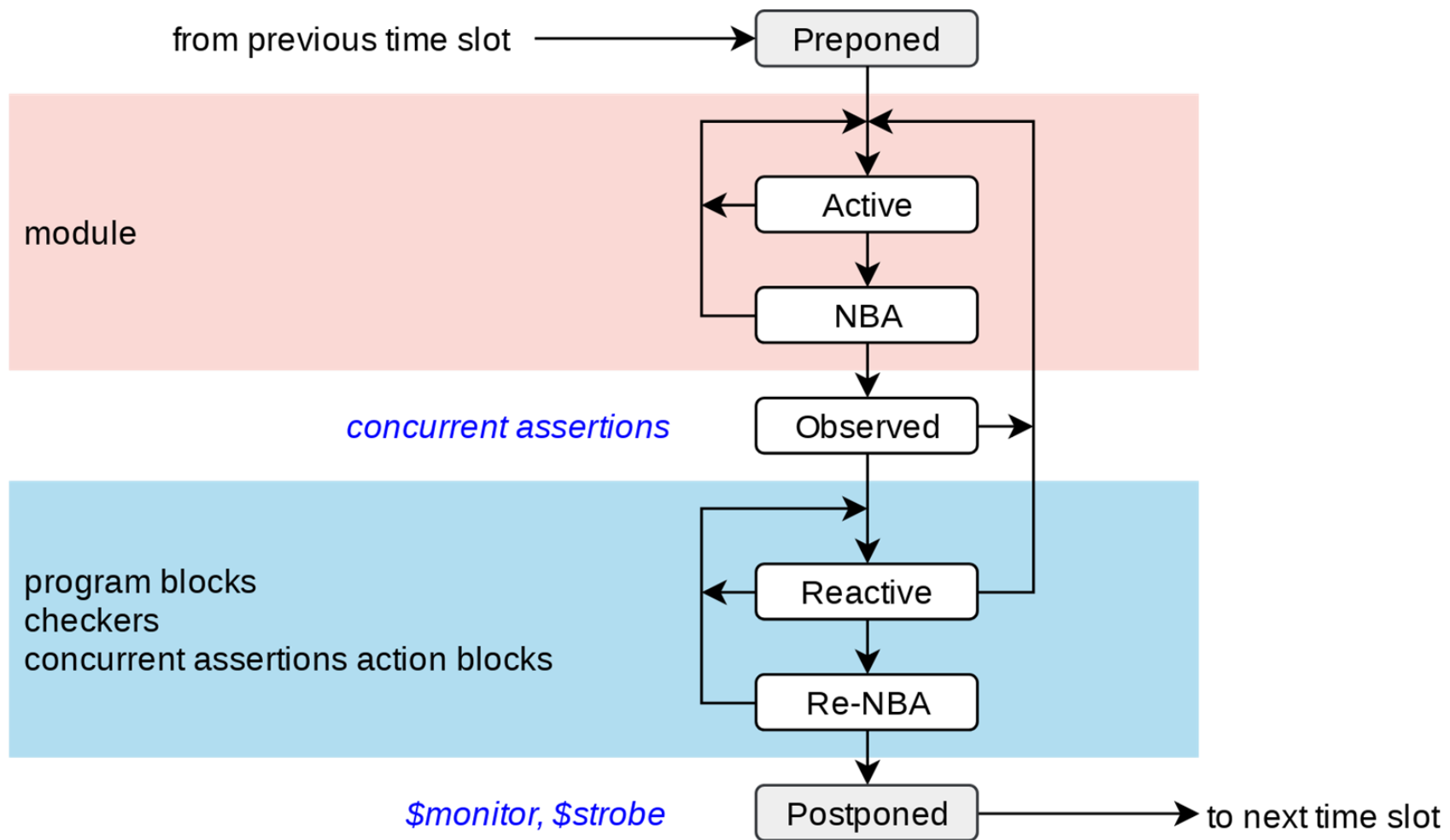
WAIT succeeded

```
begin
  a = 25;
  #20 a <= 26;
  wait(a==26);
  b = 0;
end
```

Полезное применение WAIT

```
// some other process in design
// ..
// ->event_id;

// Your process that is needed to be suspended till event is triggered
//
wait(event_id.triggered)
//
```



Спасибо за внимание

danil.bychkov@gmail.com



DISCOVER.
DESIGN.
DEVELOP.

yadro.com

Генеральный партнер конференции FPGA-Systems 2021.2

tech@exponenta.ru
exponenta.ru



ЭКСПОНЕНТА
ЦЕНТР ИНЖЕНЕРНЫХ ТЕХНОЛОГИЙ
И МОДЕЛИРОВАНИЯ

- **Технические консультации**
- **Подбор инструментов**
- **Обучение специалистов**
- **Работа на заказ**



Генеральный партнёр конференции FPGA-Systems 2021.2



Первая современная отечественная САПР, реализующая сквозной цикл проектирования печатных плат



www.eremex.ru

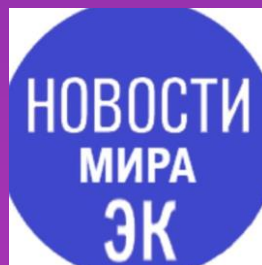
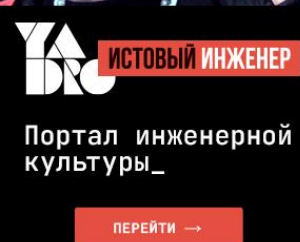
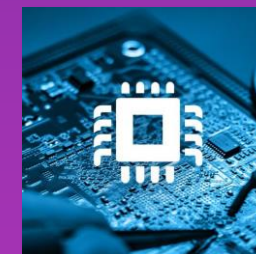
Информационные партнеры



Сообщество
приборостроителей



Русская электроника



PCB&IC SOFTWARE
PCB&IC SOFTWARE

Где найти FPGA комьюнити?



fpga-systems.ru



t.me/fpgasystems <=> @fpgasystems



youtube.com/c/fpgasystems



admin@fpga-systems.ru

