# Person GRA PORJECT

**GROUP 174** 

### ÜBERSICHT

**O1** PROBLEMSTELLUNG

**02** SIMULATIONSANSATZ

**03** IMPLEMENTIERUNG

**04** BEWERTUNG

#### PROBLEM-STELLUNG

Praktische Nutzung des TLBs

1-Beschleunigung von Speicherzugriffen (Caching)

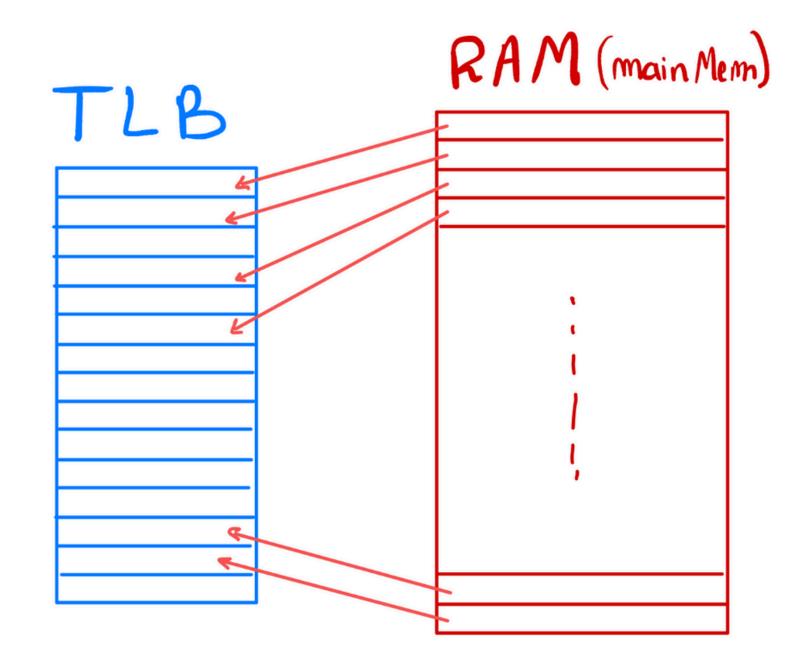
2-Effizienzsteigerung in Mehrbenutzersystemen (Isolation und Sicherheit )

**3-Verbesserte Systemleistung** 

#### PROBLEM-STELLUNG

**TLB-Eigenschaften** 

#### **Direct Mapped TLB**



#### PROBLEM-STELLUNG

**TLB-Eigenschaften** 

1-Beschleunigung von Speicherzugriffen

2-Effizienzsteigerung in Mehrbenutzersystemen

3-Verbesserte Systemleistung

### SIMULATIONSANSATZ UND IMPLEMENTIERUNG

#### SIMULATIONS-ANSATZ

Wie wird das TLB-Verhalten nachgebildet?

- Nutzung maximaler Hardwarekomponenten

- Beginn mit einem Hardwaredesign (Logisim evolution) zur Minimierung der Softwareabstraktion

- Geringerer Einsatz von Datenstrukturen

#### SIMULATIONS-ANSATZ

Wie es auf reale Situationen anwendbar gemacht wird?

-Integration des virtuellen Speichers

-Virtuelle Lookup Table

-Kommunikation mit einem virtuellen Prozessor

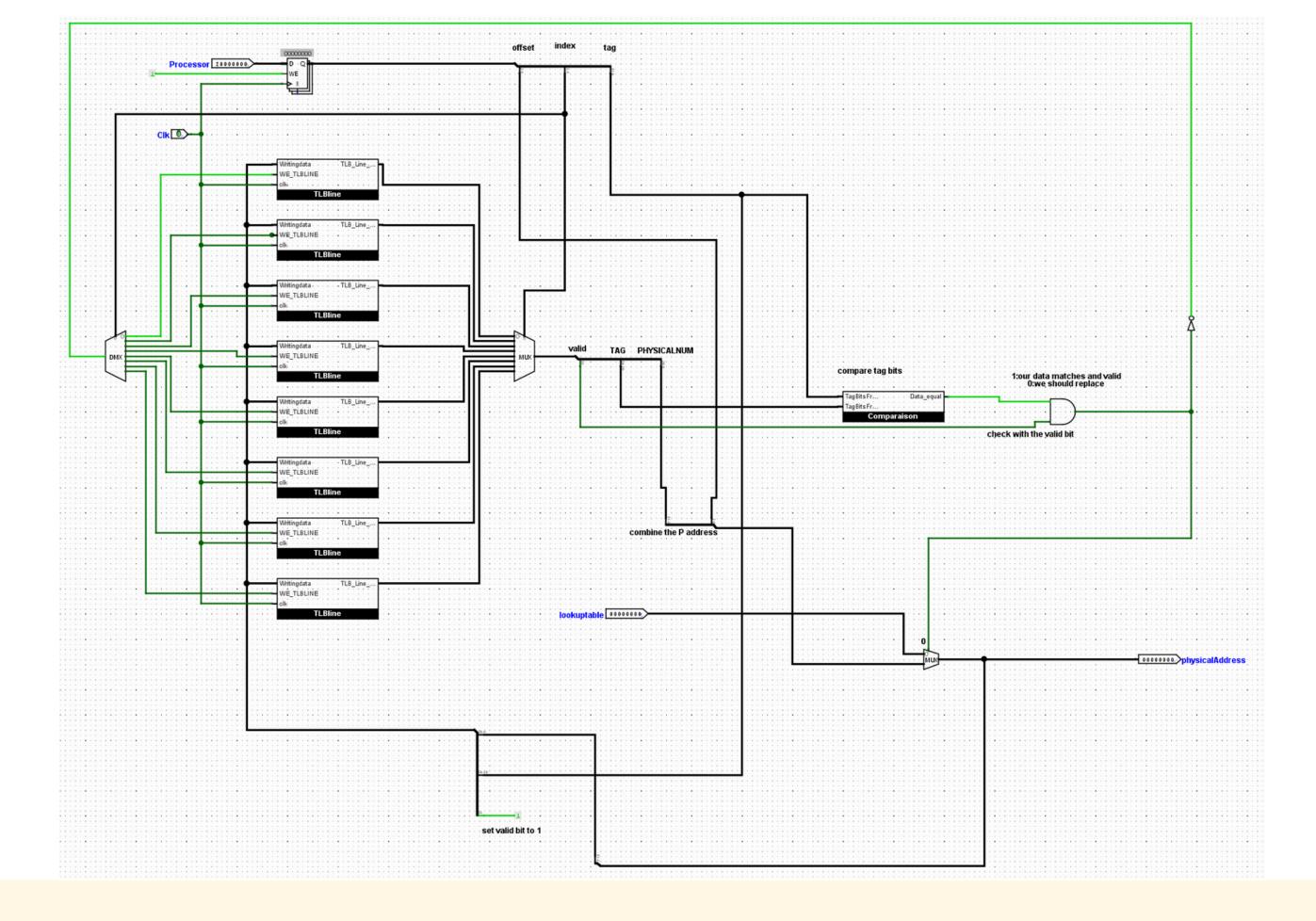
-Abdeckung von Randfällen

#### IMPLEMENTIERUNG

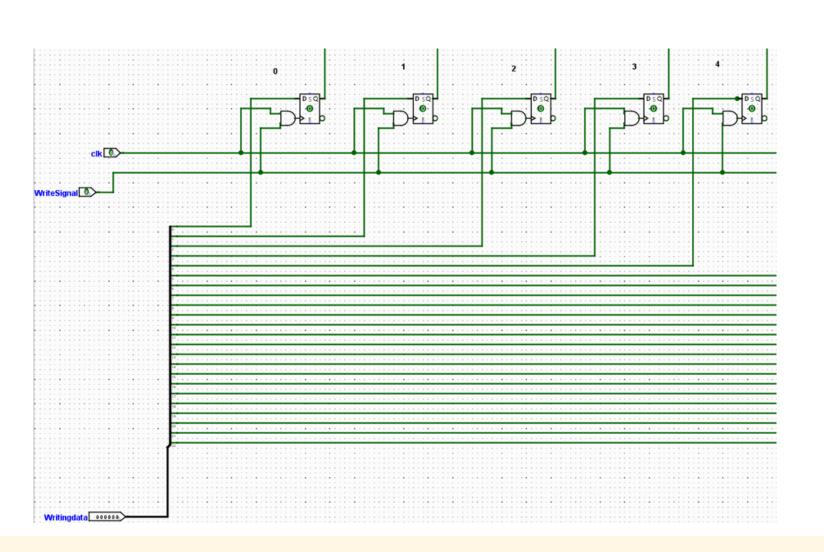
Rahmenprogramm

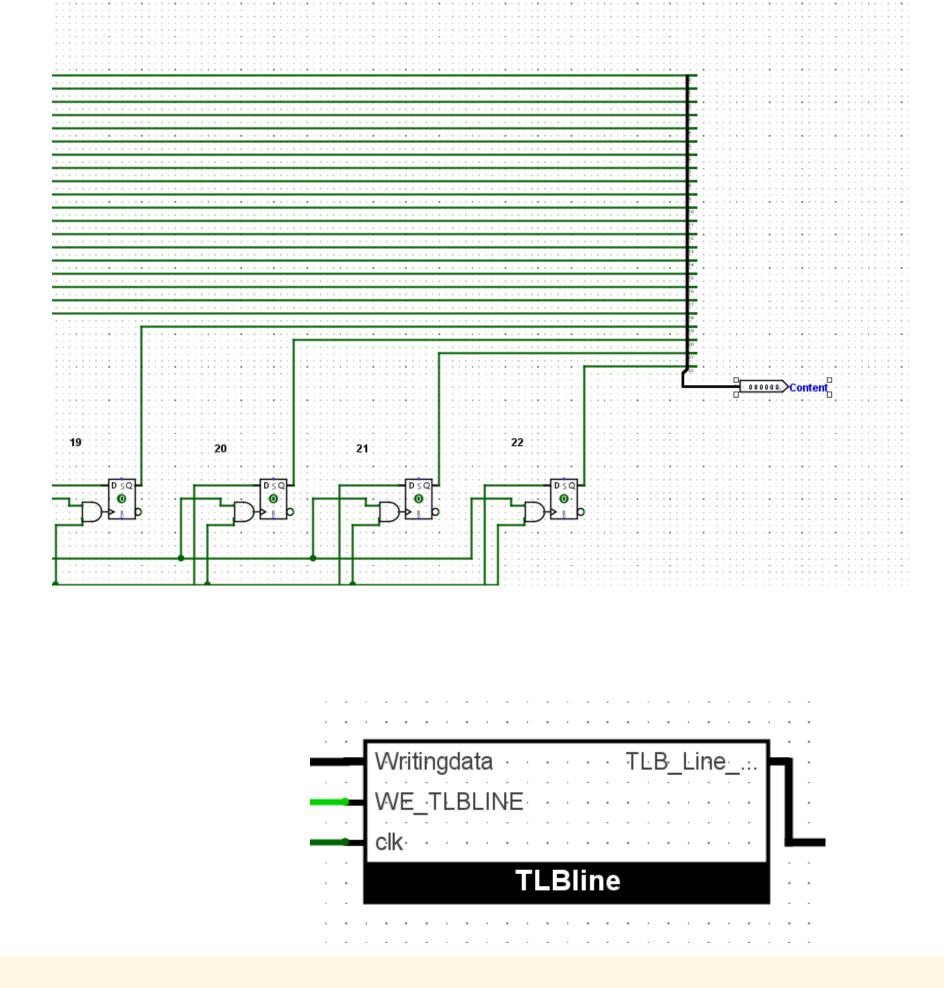
-Benutzerfreundlichkeit

-Sinnvolle Rückmeldungen und Kommentare



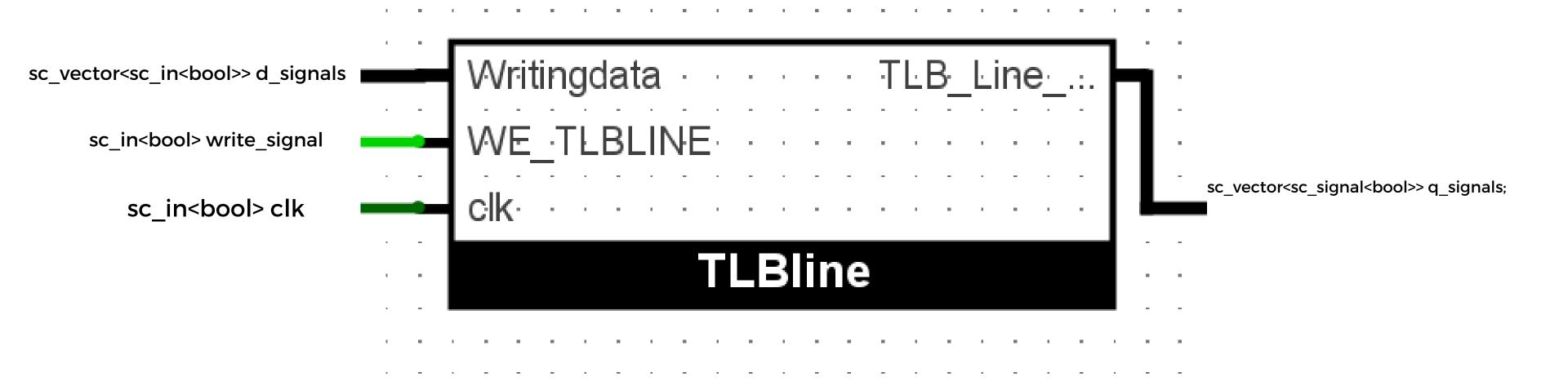
#### VON DER HARDWARE ZU .HPP-MODULEN

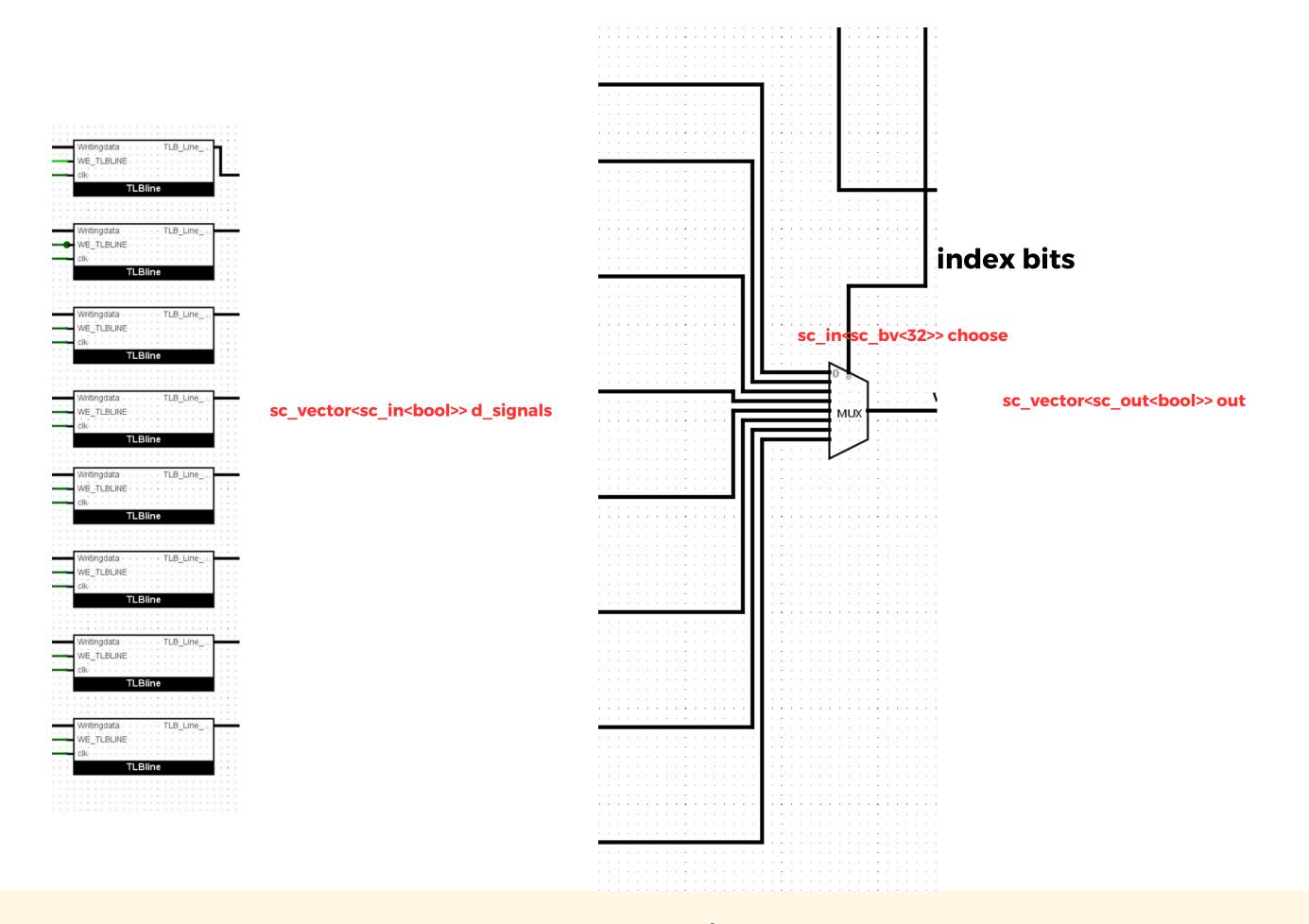




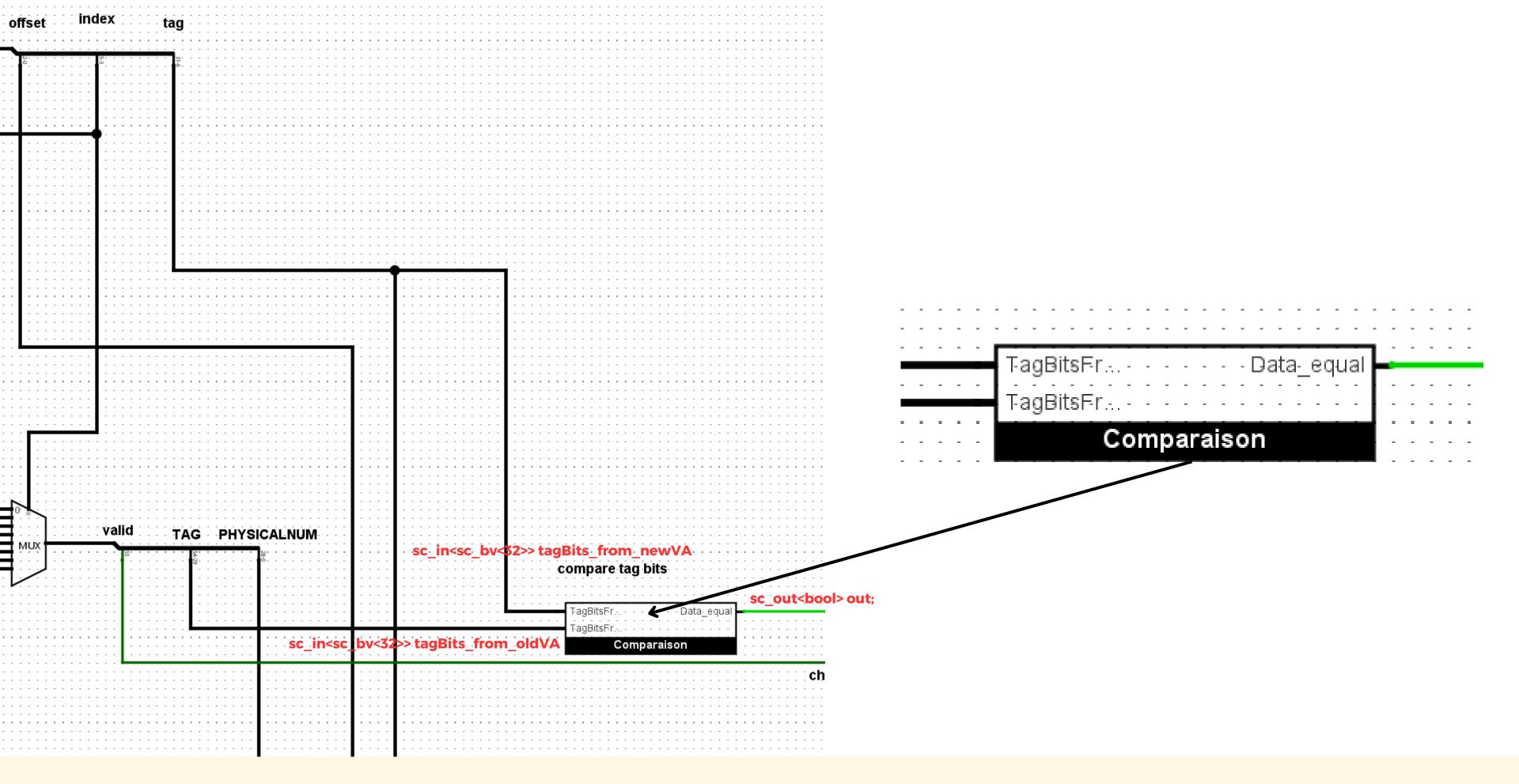
**TLB line** 

sc\_vector<sc\_signal<bool>> q\_bar\_signals
sc\_vector<D\_FLIP\_FLOP> dflipflops
sc\_signal<bool> tlb\_line\_clk
AND\_GATE and\_gate

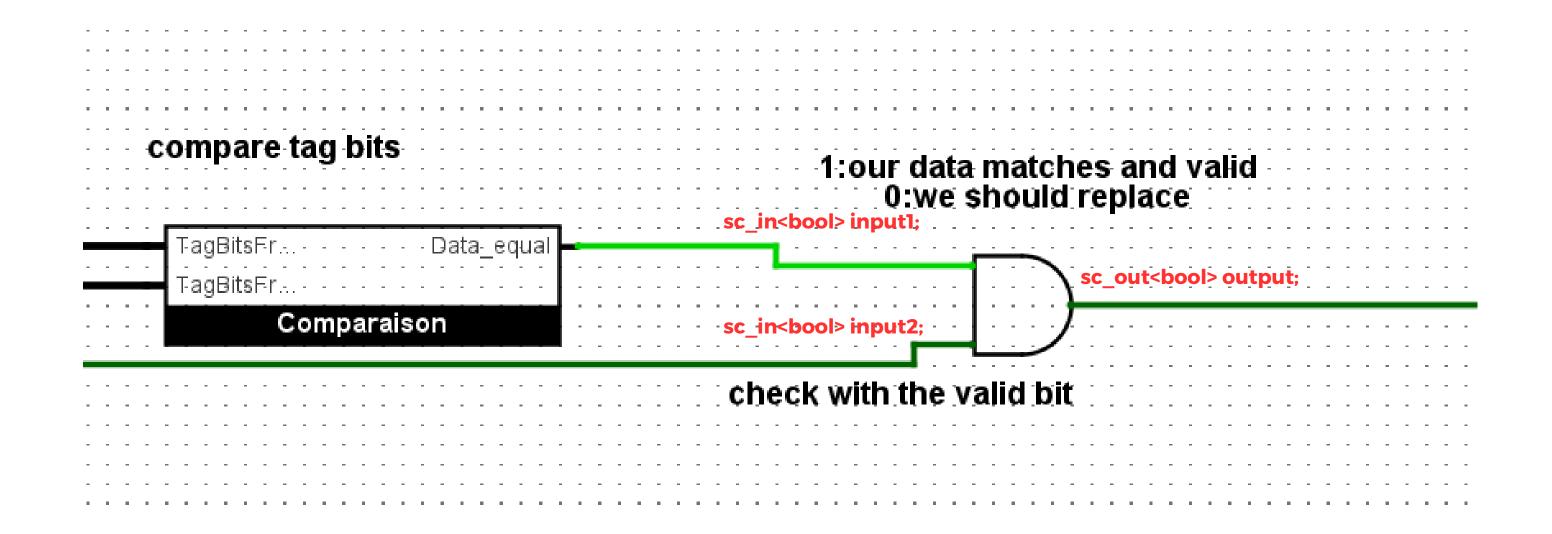


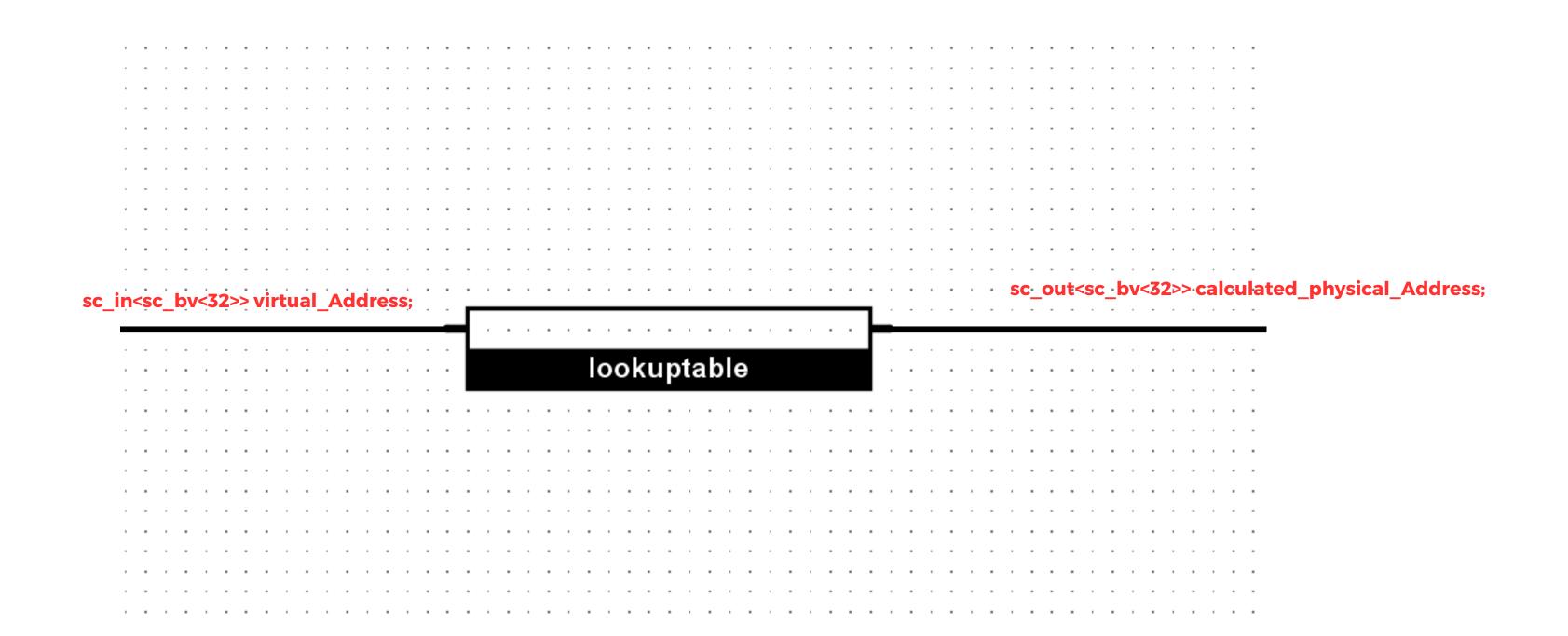


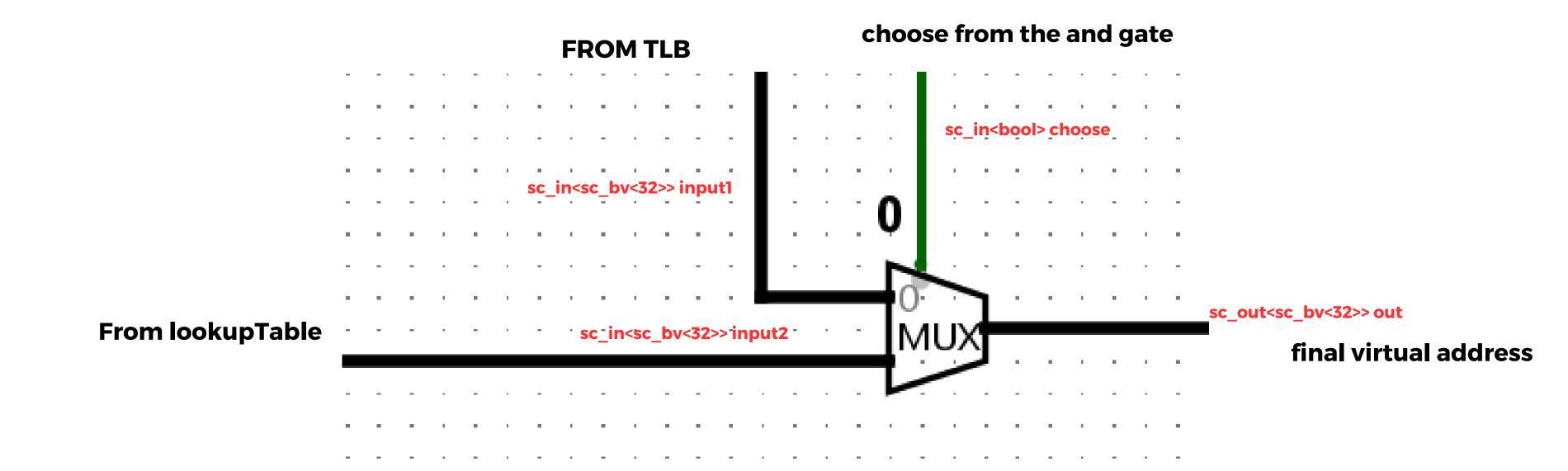
Multiplexer

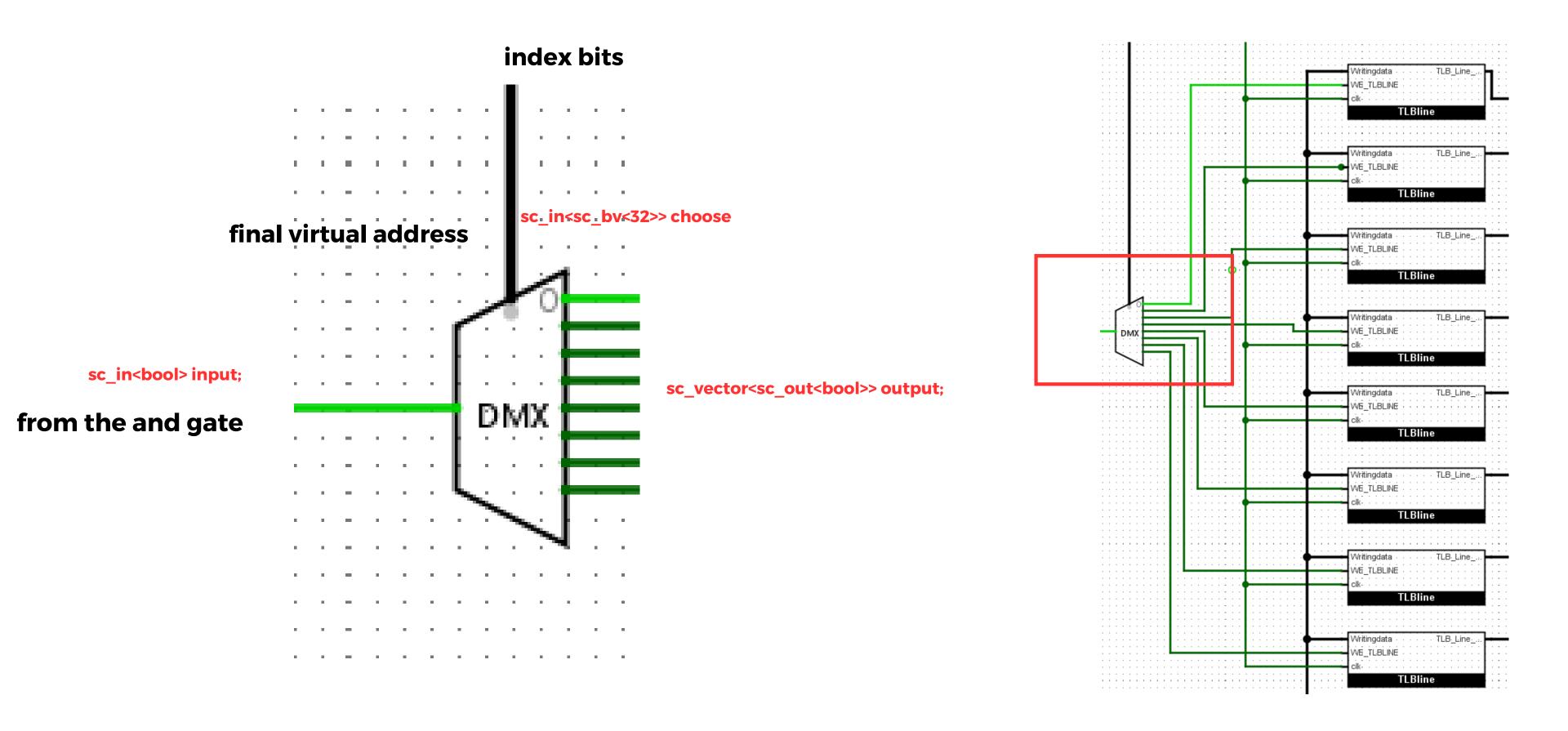


Comparator

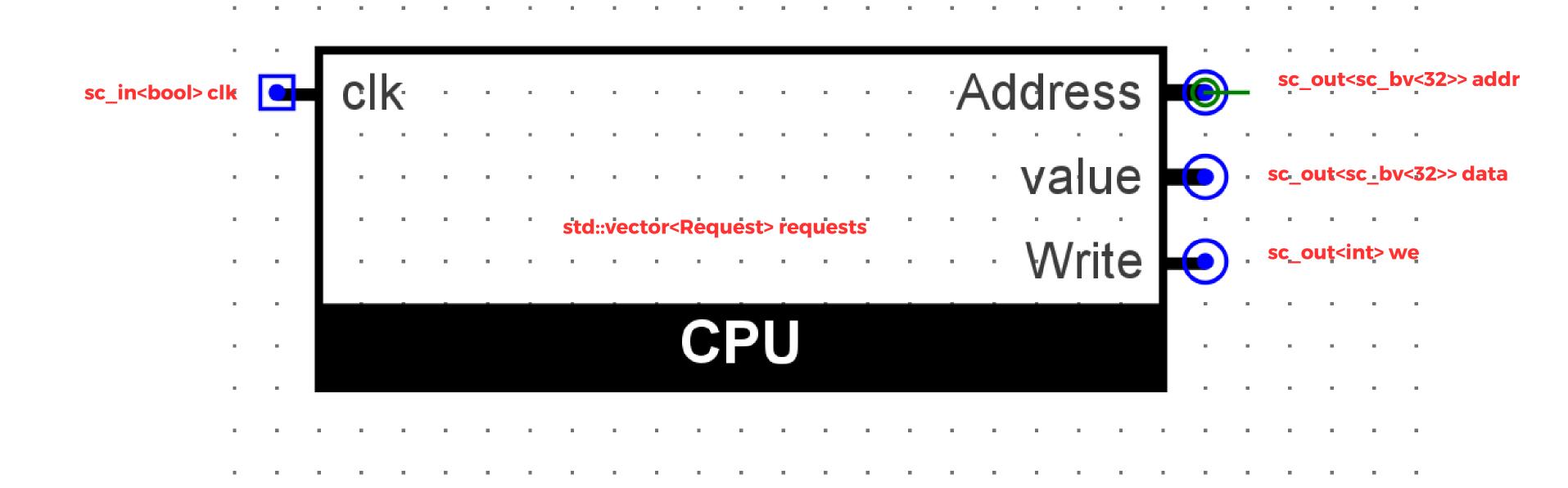


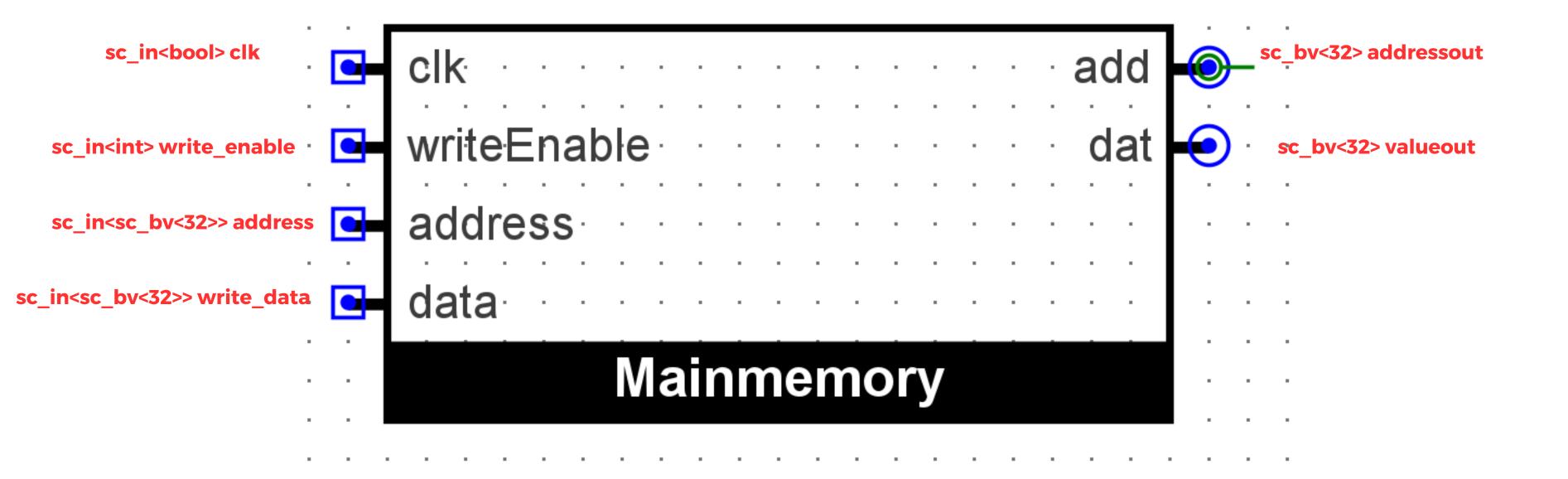


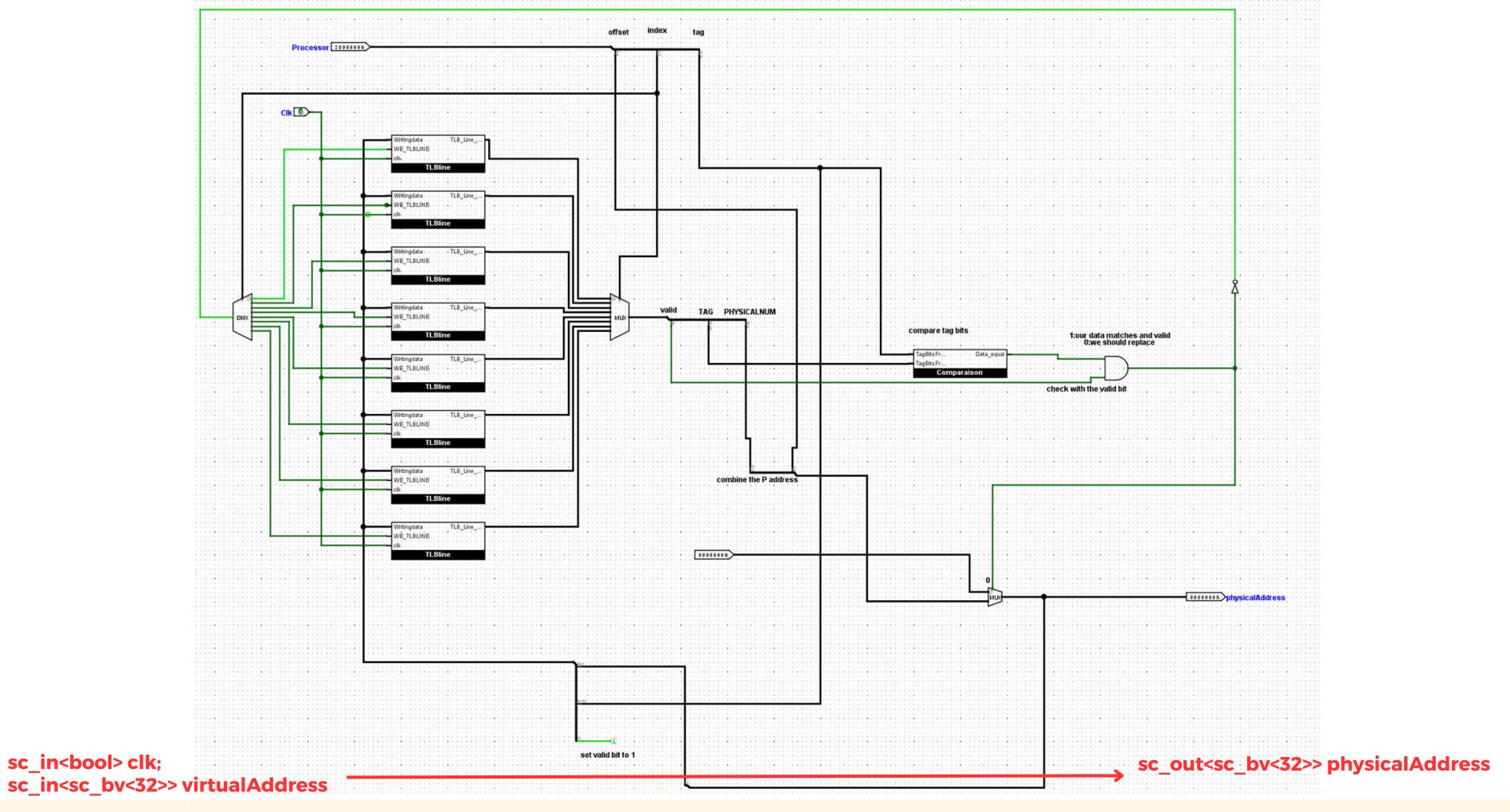




Demultiplexer







TLB mit allen Verbindungen

**Globale Konstanten** 

```
extern size_t misses;
extern size_t numRequests;
extern size_t cycle_counter;
extern int max_cycles;
extern int tlb_line_length;
extern unsigned tlbSize;
extern unsigned blocksize;
extern unsigned tlbsLatency;
extern unsigned memoryLatency;
extern unsigned v2bBlockOffset;
extern unsigned number_of_tagBits;
extern unsigned number_of_tlb_indexBits;
extern unsigned number_of_tlb_indexBits;
```

Namespaces :Namenskonflikte zu vermeiden und Code zu organisieren

**Latenz Berechnung** 

Hit latency= TLB latency+ memory latency

Hit latency= TLB latency+ 2\*memory latency

Miss penalty= memory latency

primitiv Gatteranzahl

DeMultiplexer= log\_2(TLB\_SIZE) + 2 \* TLB\_SIZE

D FlipFlop= (TLB\_LINE\_LENGTH \* TLB\_SIZE) \* 4

Multiplexer= log\_2(TLB\_SIZE) + 64TLB\_SIZE - 32

2nd MuxItiplexer= 97 gates

And Gate= 2

Not Gate = 1

Comparator = 63

Optimierungen

Page zu Page-Zuordnung:

Nutzung einer TLB-Entry für mehrere Adressen oder Adressgruppen.

Verwendung dynamischer Vektoren:

Erreichung der Flexibilität von Hardwarekomponenten.

#### Beispiel

das Speicherzugriffsverhalten einer Summe über einer verketteten Liste.

TLB-Größe	Misses	Hits	Cycles
8	16	12	496
16	8	24	464
32	8	24	464

#### KORREKTHEIT

### **Testing Units**



#### KORREKTHEIT

### DANKE FÜR IHRE AUFMERKSAMKEIT