



TECHNISCHE  
UNIVERSITÄT  
DRESDEN

Zwischenvortrag zur Studienarbeit

# Implementierung von Standarddatenstrukturen in VHDL

Jens Voß

Dresden, 01.12.2015



DRESDEN  
concept  
Exzellenz aus  
Wissenschaft  
und Kultur

## Gliederung

1. Einführung
2. Datenstrukturen
3. Aktueller Stand
4. Ausblick

## Einführung

- **Problemstellung und Motivation:**
  - für Softwareentwurf umfangreiche Bibliotheken vorhanden
  - im Hardwareentwurf ebenfalls Verwendung von Standardkomponenten
  - allerdings:
    - oft sehr speziell angepasst
    - oder sehr simpel
  - Ergänzen des Angebots der PoC-Bibliothek

## Einführung

- Aufgabenstellung:
  1. Literaturstudium zu Standarddatenstrukturen
  2. Bewertung dieser Datenstrukturen
  3. Implementierung in VHDL
  4. Test und Bewertung

# Datenstrukturen

## Stack:

- eine der einfachsten Datenstrukturen
- arbeitet nach dem LIFO-Prinzip
- geplante Funktionen:
  - push
  - pop
- mögliche Implementierungen:
  - mit RAM und Stackpointer
  - als Schieberegister

# Datenstrukturen

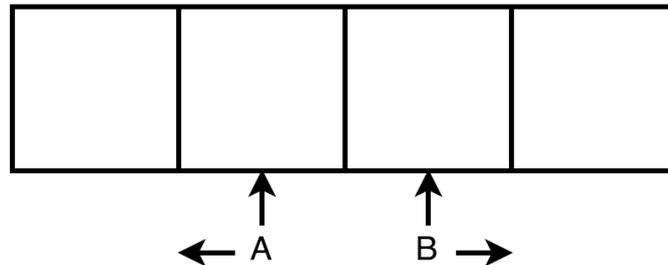
## Deque:

- double ended Queue
- beide Enden arbeiten als Stack
- kann als FIFO benutzt werden
- geplante Funktionen:
  - pushA/pushB
  - popA/popB

## Datenstrukturen

### Deque:

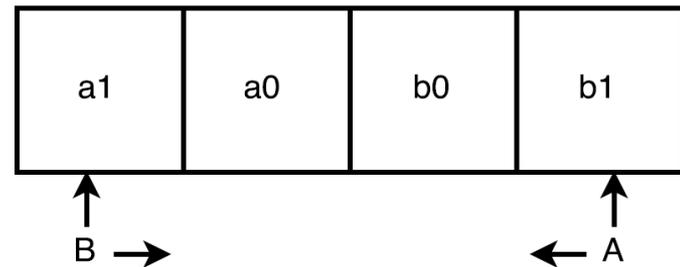
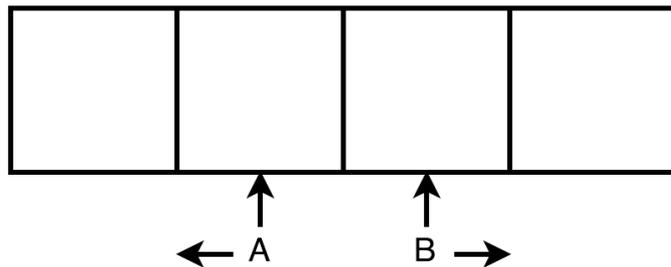
- aus den push/pop Operationen lassen sich 9 Fälle ableiten
- Überprüfung ob einfügen/löschen gültig
- Anhaltspunkt: Differenz der beiden Zeiger
  - 4 mögliche Ergebnisse



# Datenstrukturen

## Deque:

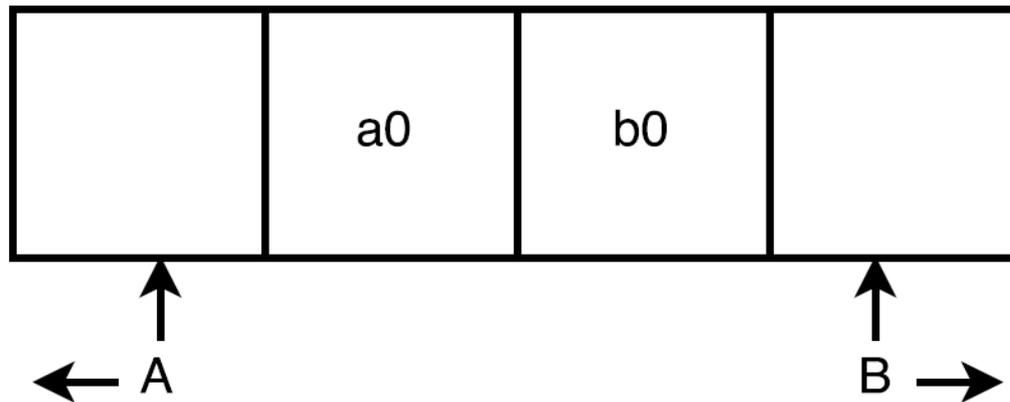
- für voll/leer:  $B - A = 1$ 
  - Auswertung mittels Flags



# Datenstrukturen

## Deque:

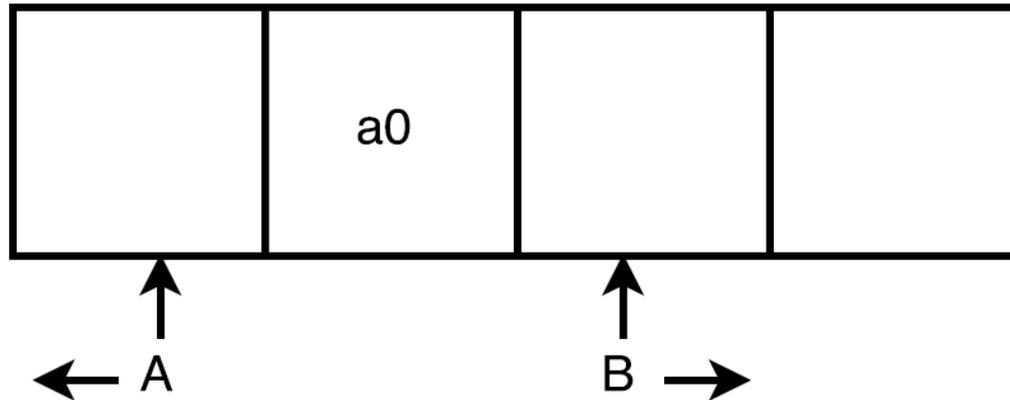
- beide dürfen schreiben/lesen:  $B - A \geq 3$



# Datenstrukturen

## Deque:

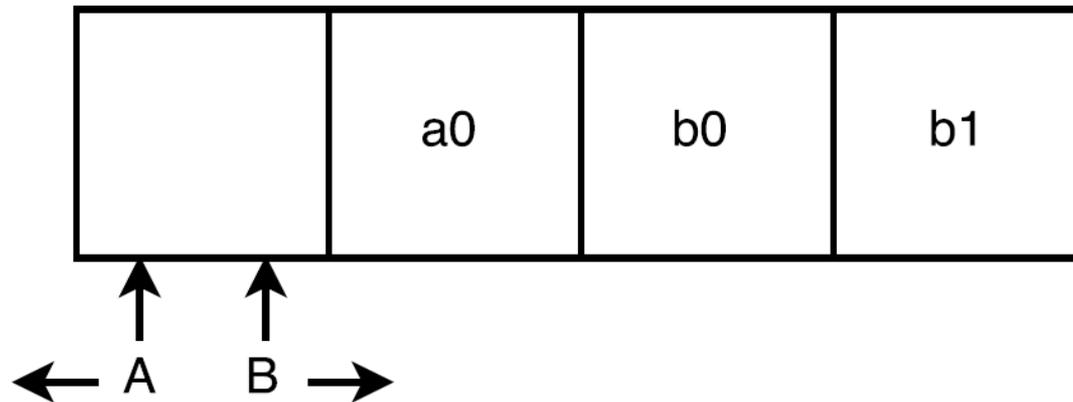
- wer darf lesen:  $B - A = 2$



## Datenstrukturen

### Deque:

- wer darf schreiben:  $B - A = 0$



## Datenstrukturen

### FIFO mit Löschoption:

- FIFO mit der Möglichkeit Elemente zu löschen
- beim Einfügen wird TAG ausgegeben
- dient zur Identifikation
- beim Löschen wird Element für ungültig erklärt
  - wird beim Lesen einfach übersprungen

# Datenstrukturen

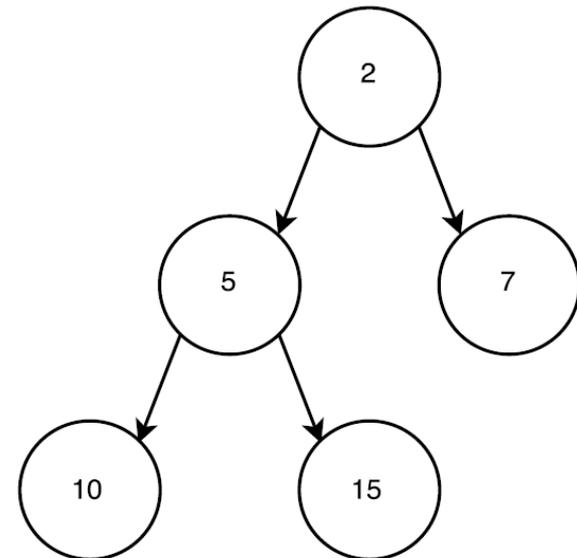
## Priority Queue:

- Elementen wird Priorität zugeordnet
- Abarbeitung entsprechend der Prioritäten
- kann als normale FIFO benutzt werden
- grundlegende Datenstruktur: Heap

## Datenstrukturen

### Heap:

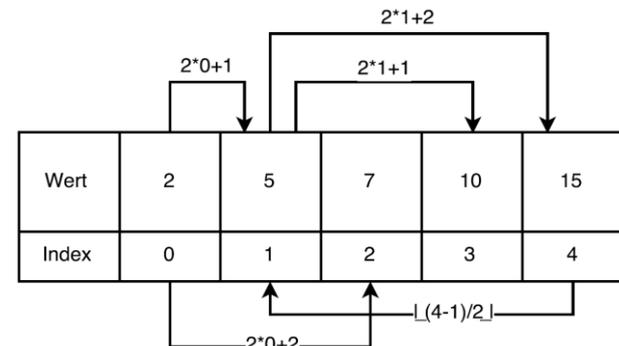
- balancierter bin. Baum
- Heap-Eigenschaft:  
 $key(i) \geq key(parent(i))$
- vollständiger Baum
  - von links einfügen
- evtl. umsordieren notwendig
- kleinster Wert immer oberster Wurzelknoten



# Datenstrukturen

## Heap:

- einfach als Array darstellbar
- Positionen:
  - Wurzel:  $\lfloor \frac{i-1}{2} \rfloor$
  - linker Knoten:  $2i + 1$
  - rechter Knoten:  $2i + 2$



# Datenstrukturen

## Priority Queue:

- Komplexitäten:
  - Einfügen:  $O(1)$
  - Löschen:  $O(1)$
  - Umsortieren:  $O(\log(n))$
  - Aufbauen des Heaps:  $O(n)$

# Datenstrukturen

## zurückgestellte Datenstrukturen:

- (doppelt) verkettete Liste
- Map
- Set
- weitere Bäume

## Aktueller Stand

### Literaturstudium:

- soweit abgeschlossen

### Stack:

- Implementierung abgeschlossen
- erste Analyse der Synthesergebnisse

### Deque:

- Implementierung begonnen

## Ausblick

### TODO:

- Stack:
  - ausführlichere Tests
  - genauere Analyse der Ergebnisse
- restliche Datenstrukturen:
  - Implementierungen abschließen
  - Analyse der Ergebnisse
- Dokumentation und Ausarbeitung vervollständigen

## Quellenverzeichnis

- [1] Martin Dietzfelbinger, Kurt Mehlhorn und Peter Sanders. *Algorithmen und Datenstrukturen / Die Grundwerkzeuge*. Springer Vieweg: Springer Vieweg, 2014. ISBN: 9783642054723.  
URL: <http://wwwdb.dbod.de/login?url=http://dx.doi.org/10.1007/978-3-642-05472-3>.
- [2] Standard Edition 7 API Specification Java™ Platform. *Deque*.  
URL: <http://docs.oracle.com/javase/7/docs/api/java/util/Deque.html>.
- [3] Standard Edition 7 API Specification Java™ Platform. *List*.  
URL: <http://docs.oracle.com/javase/7/docs/api/java/util/List.html>.
- [4] Standard Edition 7 API Specification Java™ Platform. *Map*.  
URL: <http://docs.oracle.com/javase/7/docs/api/java/util/Map.html>.
- [5] Standard Edition 7 API Specification Java™ Platform. *Outline*.  
URL: <http://docs.oracle.com/javase/7/docs/technotes/guides/collections/reference.html>.
- [6] Standard Edition 7 API Specification Java™ Platform. *Priorityqueue*.  
URL: <http://docs.oracle.com/javase/7/docs/api/java/util/PriorityQueue.html>.

## Quellenverzeichnis

- [7] Standard Edition 7 API Specification Java™ Platform. *Set*.  
URL: <http://docs.oracle.com/javase/7/docs/api/java/util/Set.html>.
- [8] Standard Edition 7 API Specification Java™ Platform. *Stack*.  
URL: <http://docs.oracle.com/javase/7/docs/api/java/util/Stack.html>.
- [9] Thomas Ottmann und Peter Widmayer. *Algorithmen und Datenstrukturen*. 5. Aufl. Spektrum Akad. Verl.: Spektrum Akad. Verl., 2012. ISBN: 9783827428035.  
URL: [http://slubdd.de/katalog?libero\\_mab215665053](http://slubdd.de/katalog?libero_mab215665053).
- [10] Gunter Saake und Kai-Uwe Sattler. *Algorithmen und Datenstrukturen / eine Einführung mit Java*. 4., überarb. Aufl. dpunkt-Verl.: dpunkt-Verl., 2010. ISBN: 9783898646635.  
URL: <http://slub.ebib.com/patron/FullRecord.aspx?p=952297>.
- [11] Robert Sedgewick und Kevin Wayne. *Algorithms*. 4. ed., 1. print. Addison-Wesley: Addison-Wesley, 2011. ISBN: 9780321573513.  
URL: <http://algs4.cs.princeton.edu/home/>.
- [12] Wikipedia. *Double-ended queue* — *Wikipedia, The Free Encyclopedia*. [Online; accessed 23-November-2015]. 2015.  
URL: [https://en.wikipedia.org/w/index.php?title=Double-ended\\_queue&oldid=690668745](https://en.wikipedia.org/w/index.php?title=Double-ended_queue&oldid=690668745).

# Quellenverzeichnis

- [13] Wikipedia. *Heap (data structure)* — *Wikipedia, The Free Encyclopedia*. [Online; accessed 24-October-2015]. 2015.  
URL: [https://en.wikipedia.org/w/index.php?title=Heap\\_\(data\\_structure\)&oldid=686781528](https://en.wikipedia.org/w/index.php?title=Heap_(data_structure)&oldid=686781528).
- [14] Wikipedia. *Priority queue* — *Wikipedia, The Free Encyclopedia*. [Online; accessed 23-November-2015]. 2015.  
URL: [https://en.wikipedia.org/w/index.php?title=Priority\\_queue&oldid=684794242](https://en.wikipedia.org/w/index.php?title=Priority_queue&oldid=684794242).



**»Wissen schafft Brücken.«**