Design- Branch History Table (BHT)

2-bit Predictor, Saturation up-down counter

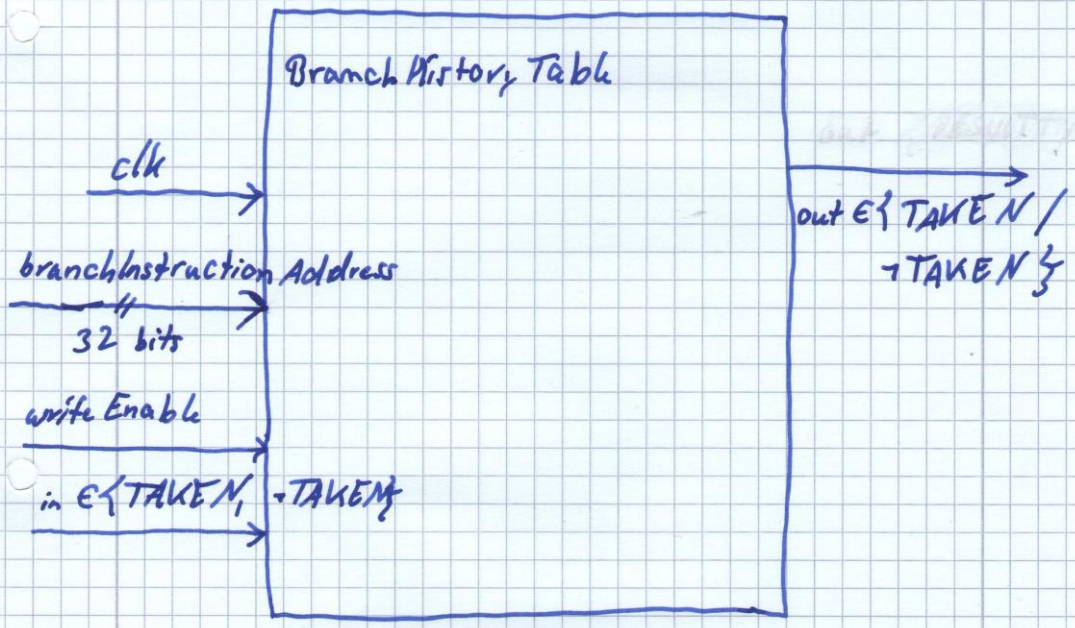
- Zwei Bits repräsentieren den aktuellen Zustand eines bedingten Sprungbefehls für jeden Eintrag in der Sprungtabelle:

Two Bits	**State**
„11“	Strongly Taken
„10“	Weakly Taken
„01“	Weakly Not Taken
„00“	Strongly Not Taken

- Anfangszustand jedes Eintrages ist Weakly Taken (siehe Seite 42)
- Neue Datentypen:
  - **PredictionState**: {STRONGLY_TAKEN, WEAKLY_TAKEN, WEAKLY_NOT_TAKEN, STRONGLY_NOT_TAKEN}
  - **Prediction**: {TAKEN, NOT_TAKEN}
- Beispiel einer BHT
  - 32 Einträge → $\log_2(32) = 5$
  - Ein Indexvektor besitzt daher die Länge n=5.

Index	*Two bits representing the state*
00000	00
00001	10
00010	10
00011	00
00100	11
00101	01
…	…
11111	11

- Register Files: Im Projektordner hdl gibt es die Entity regfile, die in der Datei regfile.vhd implementiert ist. Wie kann diese am besten eingesetzt werden?
  - Über ein write enable Signal we3 wird das Schreiben gesteuert. Nur wenn dieses Signal aktiviert ist, dann wird geschrieben. Das Signal wa3 gibt die Adresse (Index) an, welcher Eintrag geändert werden soll. Das Signal wd3 enthält das neue Wort, welches ins Register geschrieben wird.
  - Wozu gibt es jeweils zwei rd und ra Signale?



Generic Variables:

BHT_ENTRIES : INTEGER := 32

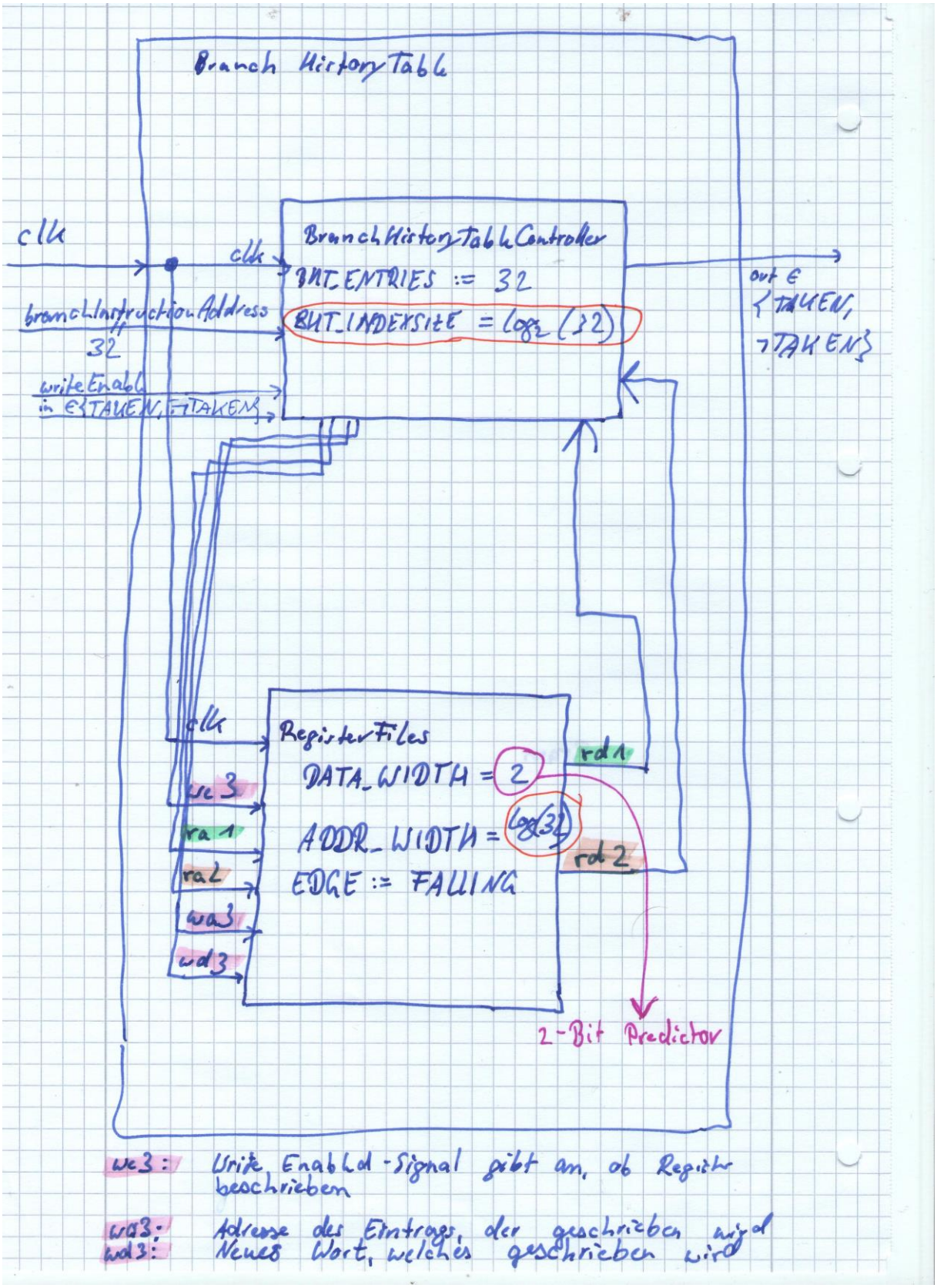
BHT_INDEXSIZE : INTEGER := $\log_2$(BHT_ENTRIES)

Signals:

writeEnable : STD_LOGIC

Die entsprechende Zeile in der Tabelle wird nur beschrieben, wenn das Signal writeEnable = '1' ist. Die Zeile wird bzgl. der in-Variable neu geschrieben.

Die entsprechende Zeile in der Tabelle, die durch die angelegte Adresse bestimmt wird, wird nicht geschrieben.



we3: Write Enable-Signal gibt an, ob Register beschrieben

wa3: Adresse des Eintrags, der geschrieben wird

wd3: Neues Wort, welches geschrieben wird