

## Dokumentation der Bauteilbibliothek für den Entwurf von Datenpfaden

Bauteil **adder4** : 4-Bit Addierer, ohne Carry-In und Carry-Out

Bauteil **inverter4**: 4-Bit Inverter

Bauteil **const0000**: 4-Bit Konstante mit dem Ausgang "0000"

Bauteil **const0001**: 4-Bit-Konstante mit dem Ausgang "0001"

Bauteile **memAxB**: PROM-Speicher der Größe *A* Wörter mit jeweils *B* Bits. Erwartet Adresse am Eingang *Adr* und liefert die dazugehörige Speicherstelle am Ausgang *Data*. Die Daten müssen in einem Textfile stehen, das im Arbeitsverzeichnis liegt. Pro Zeile in dem Textfile müssen die einzelnen Bits eines Speicherwortes ohne Leerzeichen zwischen den Bits stehen, ohne von Leerzeichen umschlossen zu werden. Die erste Zeile im Textfile wird an die Adresse 0 im PROM gelegt, die zweite Zeile an die Adresse 1 usw. Der Name der Textdatei kann im SchematicEditor von Xilinx eingegeben werden, indem das Attribut *FileName* mit dem Namen der Datei befüllt wird (Doppelklick auf das Bauteil im SchematicAdder öffnet einen Dialog mit den Attributen). Beispielzeile für ein mem16x8: 00010111

Bauteil **mux2x4**: 2-auf-1 Multiplexer, der jeweils 4-Bit breite Busse auf den Ausgang multiplext

Bauteil **mux4x4**: 4-auf-1 Multiplexer, der jeweils 4-Bit breite Busse auf den Ausgang multiplext

Bauteile **regX**: X-Bit breites Register, flankengesteuert (steigende Flanke auf Eingang *CLK*)

Bauteile **regXWE**: X-Bit breites Register mit WriteEnable ('1' auf *WE* erlaubt Schreiben, '0' verhindert Schreiben), flankengesteuert (steigende Flanke auf Eingang *CLK*)

Bauteile **regXWRWE**: X-Bit breites Register mit WriteEnable ('1' auf *WE* erlaubt Schreiben, '0' verhindert Schreiben) und Reset ('1' auf Eingang *RESET* beschreibt das Register asynchron mit dem Wert "0000", '0' bedeutet normaler getakteter Betrieb), flankengesteuert (steigende Flanke auf Eingang *CLK*)

Bauteil **regAdd4**: 4-Bit breites Register mit inkrementierendem Eingang (Werte, die am 4-Bit breiten Dateneingang *DataIn* liegen, werden um 1 erhöht, bevor sie in das Register geschrieben werden) und WriteEnable ('1' auf *WE* erlaubt Schreiben, '0' verhindert Schreiben), sowie Reset ('1' auf Eingang *RESET* beschreibt das Register asynchron mit dem Wert "0000", '0' bedeutet normaler getakteter Betrieb), flankengesteuert (steigende Flanke auf Eingang *CLK*)

Bauteil **registerfile4x4RP1WP1**: Registerfile mit 4 Registern mit jeweils 4 Bit. Das Registerfile stellt einen Leseport (Registernummer über *RPAdr* einstellbar, Daten am Ausgang *DataOut* verfügbar), sowie einen Write-Port (Registernummer über *WRAdr* einstellbar, *WE*= '1' ermöglicht Schreiben, *WE*= '0' verhindert Schreiben, Daten an *DataIn* anlegen) zur Verfügung. Register Nummer 0 enthält den Wert 0 und ist nicht beschreibbar, auch nicht bei gesetztem *WE*.