

## Aufgabe 1

Ein C Compiler speichert ein Array so, dass die beiden Zuweisungen für h und b aus folgendem C Code identisch sind:

```
1 int array [512][640][800];
2 int i , j , k;
3 int a , b;
4
5 h = array [ i ][ j ][ k ];
6 b = *(array + i*640*800 + j*800 + k);
```

Es sollen alle Elemente des Arrays mit folgendem Code aufsummiert werden.

```
1 int array [512][640][800];
2 int i , j , k;
3 int summe;
4
5 summe = 0;
6 for (k=0;k<800;k++)
7     for (j=0;j <640;j++)
8         for (i=0;i <512;i++)
9             summe += array [ i ][ j ][ k ];
```

Die Rechnerarchitektur hat einen voll assoziativen Datencache mit einer Blockgröße von 32 Bytes. Insgesamt ist der Cache 1 kByte groß. Nehmen Sie an, dass nur array Daten im Cache gespeichert sind und alle anderen Variablen in Registern gehalten werden. Ein Integer kann eine 32 Bit Zahl speichern. Nehmen Sie an, dass das array cache aligned ist, d.h. das der Speicherbereich des arrays genau am Anfang eines Cacheblocks beginnt.

- Geben Sie die Cache Miss Rate für obigen Algorithmus an.
- Nehmen Sie an, dass die Anzahl der Blöcke im Cache beliebig groß werden kann, die Blockgröße jedoch gleich bleibt. Wie groß kann die Cache Hit Rate maximal werden?
- Nehmen Sie an, der Cache hat insgesamt 512 Blöcke mit einer Blockgröße von  $1600 \cdot 4$  Bytes. Wie groß ist die Cache Hit Rate?
- Ändern Sie den Algorithmus so ab, dass die Funktion gleich bleibt, die Cache Hit Rate jedoch maximal wird.
- Schlagen Sie eine alternative Blockgröße bei gleicher Gesamtgröße des Caches vor, mit der Sie die Cache Hit Rate für den modifizierten Algorithmus aus d) maximieren. Wie groß wird dann die Cache Hit Rate?

## Aufgabe 2

Für einen Algorithmus zur Positionsbestimmung müssen zwei Matrizen A und B multipliziert werden  $C = A \cdot B$ . Jedes Matrixelement ist eine 16 Bit Zahl. Die Matrixmultiplikation ist jedesmal notwendig wenn ein neuer Messwert von den Abstandssensoren vorliegt. Die Sensoren liefern Daten mit einer Rate von 10 MSamples/s. Die Werte der Elemente der Matrizen A und B ändern sich mit jedem neuen Sensorwert.

$$c_{ij} = \sum_{k=0}^2 a_{ik} \cdot b_{kj}; \quad A = \begin{pmatrix} a_{00} & a_{01} & a_{02} \\ a_{10} & a_{11} & a_{12} \\ a_{20} & a_{21} & a_{22} \end{pmatrix} \quad (1)$$

Der Algorithmus zur Matrixmultiplikation ist in Gleichung (1) angegeben. Die beiden Matrizen A und B, sowie die Ergebnismatrix C befinden sich in einem Speicher mit 16 Bit Breite. Der Speicher kann in jedem Takt ein Datum lesen oder schreiben. Sie sollen eine Schaltung zur Matrixmultiplikation entwerfen.

- a) Berechnen Sie die notwendige Taktfrequenz mit dem der Speicher getaktet werden muss um die Daten für jede Berechnung aus dem Speicher auszulesen und das Ergebnis im Speicher abzulegen. Nehmen Sie also an, dass Ihre Schaltung keine Matrixelemente zwischenspeichert.
- b) Die erforderliche Taktfrequenz für den Speicher aus der vorherigen Aufgabenstellung ist zu hoch, d.h. Sie können keinen Speicher mit einer so hohen Taktfrequenz bauen. Nehmen Sie jetzt an, dass Sie beliebig viele Matrixelemente in ihrer Schaltung für die Berechnung zwischenspeichern können. Wie hoch ist dann die minimal mögliche Speicherfrequenz?
- c) Um den Speichertakt zu reduzieren sollen in drei 16 Bit Registern r0, r1, r2 die drei Elemente einer Zeile der A Matrix zwischengespeichert werden. Das Timing der Eingangsdaten ist in Abbildung 1 dargestellt.

Zusammen mit dem Element  $a_{00}$  ist das Signal "start" aktiv. Die ersten drei Daten sind die erste Zeile der A Matrix. Dann folgt die erste Spalte der B Matrix, dann die zweite Spalte der B Matrix und dann die dritte Spalte der B Matrix. Dann folgt die zweite Zeile der A Matrix. Zeichnen Sie einen Datenpfad für die Berechnung der Matrixmultiplikation und die dazugehörigen Steuersignale. Zeichnen Sie das Timing der Steuersignale in das Timingdiagramm ein. Zeichnen Sie die Werte von wichtigen Signalen im Datenpfad in das Diagramm ein. Nehmen Sie als Werte für die Matrixelemente die Werte aus der Zeile "exdata" an. Der Ausgang ihrer Schaltung ist ein valid und ein 16 Bit cout Signal. Valid soll aktiv werden, sobald ein berechnetes Element der C Matrix am Datenausgang vorliegt.

Geben Sie in Zeile "cout" an welches Matrixelement der Matrix C am Ausgang vorliegt, also beispielsweise "c01". Geben Sie in der Zeile "cout(data)" den zugehörigen Datenwert als Zahl an.

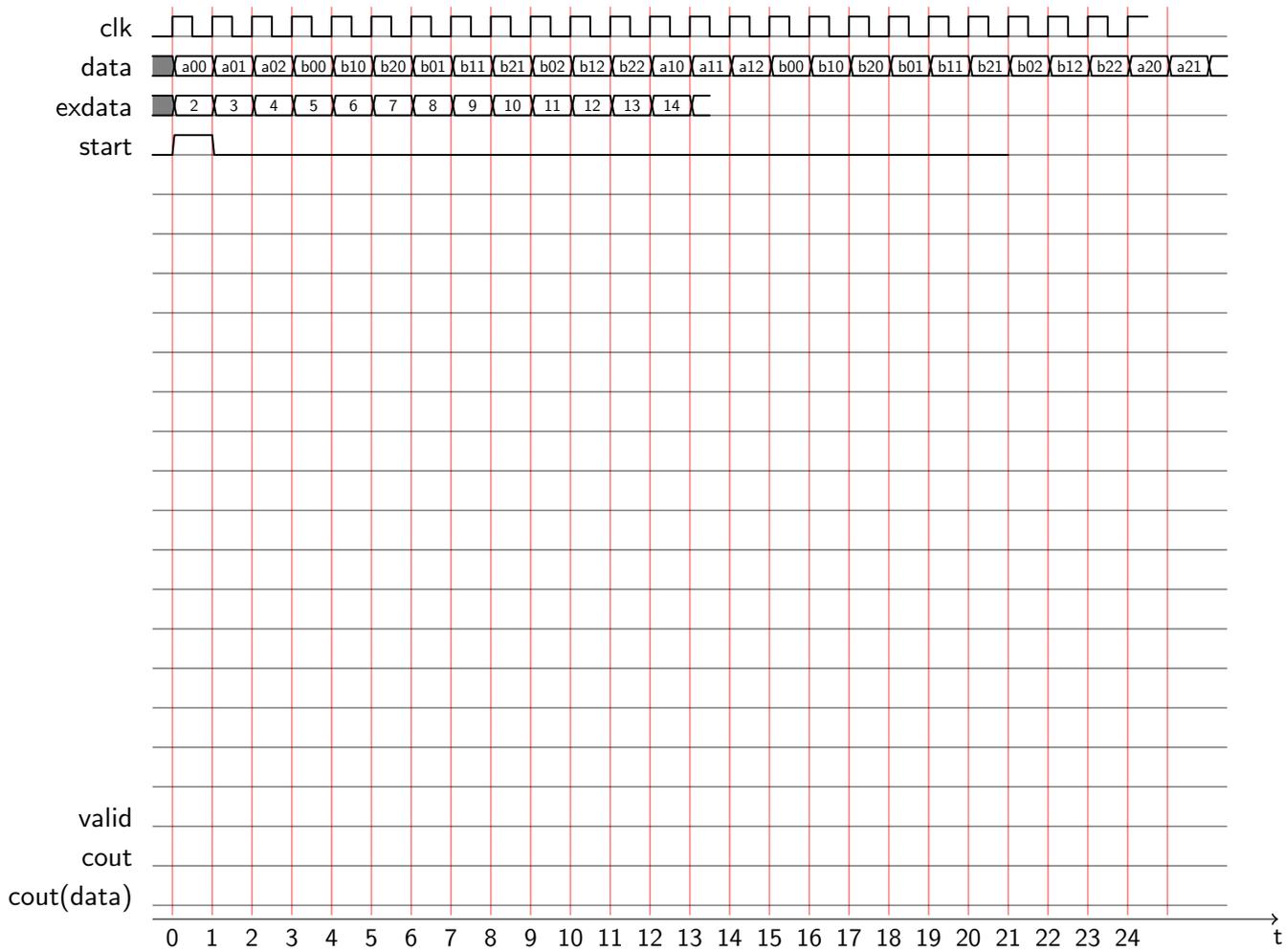


Abbildung 1: Timingdiagramm zur Matrixmultiplikation