

## Aufgabe 1)

Der Adressbus ist ein Teil eines Computers, der verwendet wird, um die physikalische Adresse von Speicherzellen oder anderen Geräten anzugeben, auf die zugegriffen werden soll. Er bestimmt, welcher Teil der Arbeitsspeicher oder welches Gerät angesprochen werden soll.

Der Datenbus ist ein Teil eines Computers, der verwendet wird, um Daten zwischen verschiedenen Komponenten des Systems zu übertragen. Er überträgt die Daten, die gespeichert oder verarbeitet werden sollen, von einer Komponente des Systems zu einer anderen.

Der Steuerbus ist ein Teil eines Computers, der verwendet wird, um Steuerbefehle zwischen verschiedenen Komponenten des Systems zu übertragen. Er überträgt Befehle, die angeben, wie Daten verarbeitet werden sollen, von einer Komponente des Systems zu einer anderen.

## Aufgabe 2)

Ein Statusregister ist ein Speicherbereich in einem Prozessor, der verwendet wird, um den aktuellen Status des Systems zu speichern. Es enthält Informationen über den Zustand des Prozessors, wie z.B. Flags für die Übertragung, die Unterbrechung, die Division durch Null und so weiter. Das Statusregister enthält auch Flags, die auf den Ergebnissen von Rechenoperationen basieren, wie z.B. das Carry-Flag (C), das anzeigt, ob eine Übertragung stattgefunden hat.

Das Statussignal c (Carry-Flag) geht vom Akkumulator zum Statusregister und gibt an, ob bei einer bestimmten Rechenoperation eine Übertragung stattgefunden hat. Dies ist wichtig, weil es dem Prozessor darüber informiert, ob er zukünftigen Operationen mit Übertragungen rechnen muss.

Die Signale des Datenbus allein reichen hier nicht aus, weil sie nur den Wert der Daten, die verarbeitet werden, transportieren. Um den Status des Prozessors abzubilden, sind zusätzliche Informationen erforderlich, wie z.B. ob es bei einer Rechenoperation eine Übertragung gab.

## Aufgabe 3)

Das Statusregister beeinflusst die Abarbeitung eines Befehlsatzes, indem es Flags für verschiedene Zustände des Prozessors speichert. Beispielsweise kann das Carry-Flag (C) anzeigen, ob bei einer Rechenoperation eine Übertragung stattgefunden hat, und dies kann Auswirkungen auf die Ausführung von Befehlen haben, die auf diesem Flag basieren, wie z.B. Additionen oder Subtraktionen.

Ein weiteres Flag kann das Zero-Flag (Z) sein, das anzeigt, ob das Ergebnis einer Berechnung Null ist. Es kann auch Auswirkungen auf die Befehlsatz haben, die auf der Nullstellung des Ergebnisses basieren wie z.B. Sprungbefehle in Verbindung mit den Vergleichsoperationen. Das Statusregister kann auch Flags erhalten, die Auswirkungen auf Interrupts haben, wie z.B. das Interrupt-Enable-Flag (IE), das anzeigt, ob Interrupts aktiviert sind.

## Aufgabe 4)

Die Ausgangssignale der Instruktionszähler (Instruction Pointer - IP) sind Signale, die von dem IP-Register ausgegeben werden und verwendet werden, um die nächste Befehlsadresse im Speicher anzugeben, die ausgeführt werden soll. In diesen Modellprozessoren, in dem sie das Programm beschreiben, wird die Ausgangssignale der Instruktionszähler verwendet, um die Adresse des nächsten Befehls anzugeben und die Ausführungsposition des Prozessors zu ändern, wenn der jump Befehl aufgerufen wird.

## Aufgabe 5)

clk → Akkumulator (clk)

brn → Instruktionszähler (n) Statusregister (n)

brc → // // (C)

brz → // // (Z)

jmp → // (S)

sub → Akkumulator (sub) (S)

add → Akkumulator (r)

sta → Ram (w)

lda → Akkumulator (ld) (r)

nop → Nop

ind → Multiplexer

### Aufgabe 6)

Das Signal "ind" wird auf einen Wert von "1" gesetzt. Das Signal "ind" verwendet wird, um anzuzeigen, dass der Adressbus des Modellrechners eine indirekte Adresse enthält. Eine indirekte Adresse ist ein Speicherplatz, der die tatsächliche Adresse enthält, auf die zu greifen verdon soll, anstatt die Adresse direkt zu verwenden. Das bedeutet, dass wenn "ind" = 1 ist, dass es darauf hinweist, dass der Adressbus mit einer indirekten Adresse beschriftet wurde.

### Aufgabe 7)

Mit dem gegebenen Programm und dem Startwert 0, die 15 Schritte durchlaufen

01 LDA #2  $\Rightarrow$  Akkumulator = 2, IP = 1  
11 STA (8)  $\Rightarrow$  Speicherplatz 8 = 2, IP = 2  
2: ADD (8)  $\Rightarrow$  Akkumulator = 4, IP = 3  
3: ADD #5  $\Rightarrow$  // = 9, IP = 4  
4: SUB #3  $\Rightarrow$  // = 6, IP = 5  
5: JMP 1  $\Rightarrow$  IP = 1

Nach 15 Schritten werden die Werte im Akkumulator 6; im Instruktionsregister 1; im Datenregister 2 und im Instruktionszähler 1 bleiben

ganz!