

Name:

Matrikelnr.:

Schreiben Sie bitte Namen und Matrikelnummer auf alle Ihre beschriebenen Blätter. Nutzen Sie auch die Rückseiten.

1. a) Wandeln Sie die folgenden Dualzahlen in Hexadezimalzahlen um. (16 Punkte)

b) Geben Sie an, ob es sich bei den folgenden Dualzahlen aus einem 4-Bit großen Zahlenraum, in dem sowohl positive als auch negative Zahlen dargestellt werden können, um eine positive oder negative Zahl handelt und bilden Sie dann jeweils die Komplementärzahl.

6) a) (8 Punkte)
 $(0101101011111111010)_2 = (5AFA)_{16}$
 $(0010101101101010)_2 = (2B6A)_{16}$

8) b) (8 Punkte)
 $1111 \Rightarrow$ Es handelt sich um eine negative Zahl; pos. Komplementärzahl: 0001
 $0111 \Rightarrow$ Es handelt sich um eine positive Zahl; neg. Komplementärzahl: 1001
 Vorzeichen

$0101 = 4 + 1 \cdot 2^0 = 5$	$0000 + 1$	-1	0
$1010 = 1 \cdot 2^3 + 2 = 10$	$0000 + 1$	-2	1
$1011 = 8 + 2 + 1 = 11$	0001	-3	2
$0110 = 4 + 2 = 6$	$1110 + 1$	-4	3
		-5	4
		-6	5
		-7	6
		-8	7

1010 A
 1101 B
 1110 C
 1111 D E F

2. Erklären Sie den Gültigkeitsbereich von Variablen. Wo kann eine Variable vereinbart werden, wo ist sie gültig, welchen Sinn haben verschiedene Gültigkeitsbereiche? (12 Punkte)

Der Gültigkeitsbereich von Variablen meint die Grenzen, in denen Variablen gültig sind, d. h. bekannt sind und einen Wert besitzen.

Es gibt 3 verschiedene Arten:
 Superglobale Variablen: werden im gesamten Skript erkannt, also innerhalb und außerhalb von Funktionen. Sie sind vom System vorgegeben, z.B. $\$_POST[]$ in PHP

globale Variablen: werden außerhalb von Funktionen definiert, sind dort gültig, können aber mithilfe der Parameterliste der Funktion übermittelt werden. Außerdem können Variablen mit der Anweisung "global" global gemacht werden und besitzen dann auch außerhalb dieser Funktion Gültigkeit

lokale Variablen: werden in einer Funktion definiert und
bestehen auch nur dort Gültigkeit.

↳ Lokale Variablen sind zum Beispiel sinnvoll, wenn
in Schleifen Wiederholungen getätigt werden sollen.

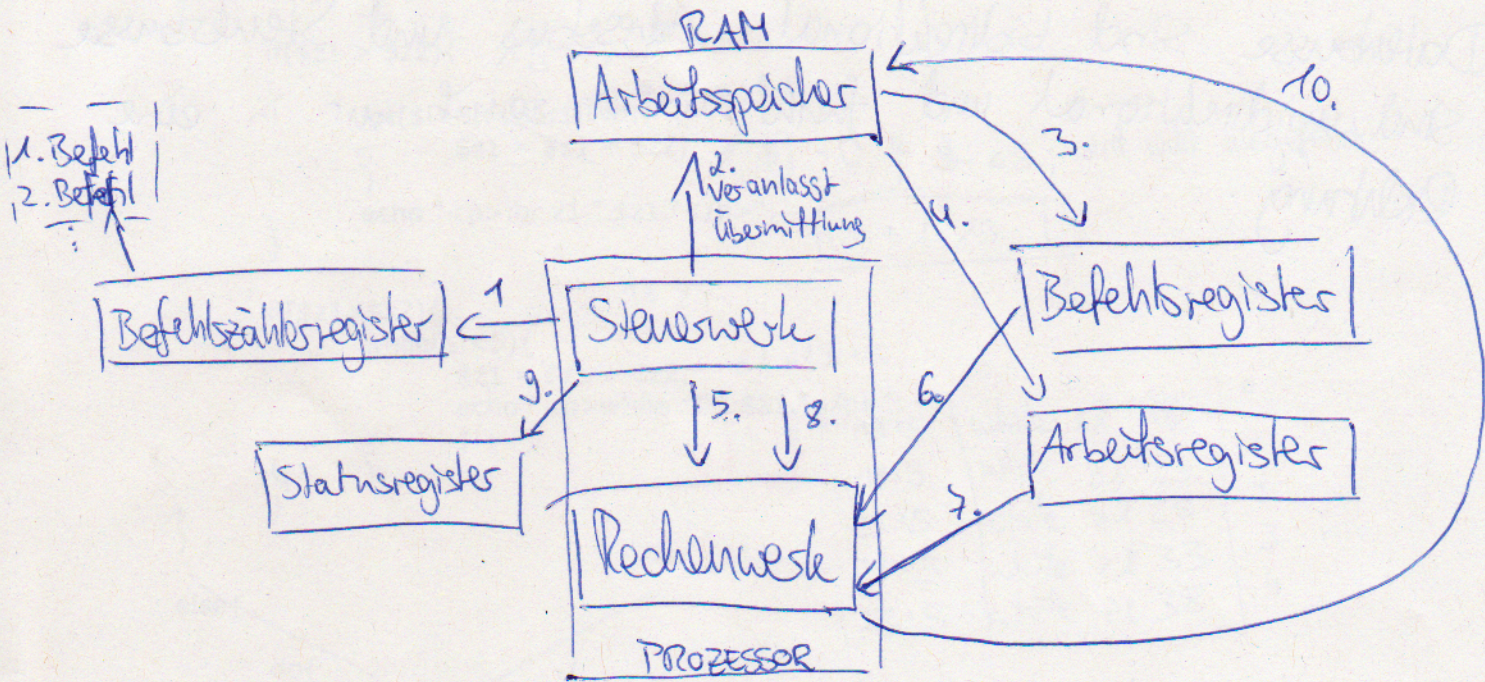
Superglobale Variablen, wenn Daten von einer
html-Seite übermittelt werden und im gesamten
PHP-Skript damit gerechnet wird.

Durch verschiedene Gültigkeitsbereiche können
verschiedene Berechnungen durchgeführt werden.
Man kann ~~das~~ sich den ~~den~~ geeigneten Gültigkeits-
bereich für eine Variable definieren.

3. Erklären Sie den Begriff „Zentraleinheit“, deren einzelne Bestandteile und ihre Zusammenarbeit. Sie können eine Zeichnung erstellen. (12 Punkte)

Die Zentraleinheit, auch CPU genannt, * mithilfe von Datenbussen besteht aus Prozessor, Arbeitsspeicher und den Anschlusssystemen.

11



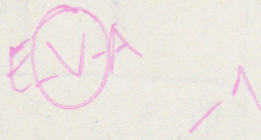
1. Das Steuerwerk beschafft sich mithilfe des unidirektionalen Adressbuses die Information Φ über den als nächstes abzuarbeitende Befehl.
2. Das Steuerwerk veranlasst, dass der Arbeitsspeicher die Operanden an das Arbeitsregister und die gebrauchten Daten an das Befehlsregister übermittelt.
- 3., 4. Arbeitsspeicher übermittelt wie in 2. beschrieben.
- 5., Steuerwerk veranlasst das Rechenwerk sich die Daten und Operanden aus Befehlsregister und Arbeitsregister* zu beschaffen und zu verarbeiten.
- 6., 7. Das Rechenwerk beschafft sich die Daten und führt die Operation durch.

8. Das Steuerwerk veranlasst das Rechenwerk,
das Ergebnis an den Arbeitsspeicher zu übermitteln.

9. Das Steuerwerk meldet den Status an das
Statusregister

10. Das Rechenwerk übermittelt das Ergebnis an das
Arbeitsregister.

Datenbusse sind bidirektional, Adressbus und Steuerbusse
sind unidirektional und funktionieren somit nur in eine
Richtung.



4. Bearbeiten Sie den folgenden Quellcode? (16 Punkte)
 Was bedeutet \$_POST[] und woher beziehen diese Variablen ihre Werte? (4 Punkte) (4)
 Welchen Gültigkeitsbereich haben die Variablen \$z1 und \$z2? (2 Punkte) (2)
 Welche Bildschirmausgabe ergibt sich aus dem Quellcode, wenn:
 \$z1 = 8 und \$z2 = 5? (10 Punkte) (10)

<?php

```
$z1 = $_POST["z1"]; 8
$z2 = $_POST["z2"]; 5
```

```
if ($z1 > $z2) {
    $z1 = $z1 + 5; 8+5 = 13
```

```
    if ($z1 > $z2) { 13 > 5
```

```
        for ($i=0; $i<5; $i++) {
```

 \$z1 = \$z1 + \$z2; // \$i = 0 & \$i < 5; wird nicht ausgeführt

```
        }
        echo "<p>for z1 ".$z1."</p>";
```

for z1 13

```
    }
    if ($z1 > $z2) { 13 > 5
```

```
        while ($i < 5) {
```

 \$z1 = \$z1 + \$z2; z1 = 13 + 5

```
        echo "<p>while z1 ".$z1."</p>";
```

```
        $i++;
```

| | | |
|------|-------------|---|
| 13+5 | while z1 18 | 0 |
| 18+5 | while z1 23 | 1 |
| 23+5 | while z1 28 | 2 |
| 28+5 | while z1 33 | 3 |
| 33+5 | while z1 38 | 4 |

```
    }
    else {
        do {
            $z2 = $z2 + $z2;
            echo "<p>do z2 ".$z2."</p>";
        } while ($z2 <= 5);
    }
}
```

?>

\$_POST ist eine superglobale Variable und ein zusammengesetztes Datentyp.
 Als assoziatives Feld, d.h. ein nicht nummerisches Feld →

--	--	--	--	--	--

 besteht es aus einem <form>-Element mit
 method=post Werte, die im HTML eingegeben worden sind.

\$z1 und \$z2 sind globale Variablen

Bildschirmausgabe: for z1 13 ✓
 while z1 18
 while z1 23
 while z1 28 ✓
 while z1 33
 while z1 38

5. Welche Bildschirmausgabe ergibt sich aus dem folgenden Quellcode? (10 Punkte)

8

<?php

```
$op = '+';
$z1 = 1;
$z2 = 1;
wahl($op, $z1, $z2);
echo $ergebnis;
```

```
function wahl($rechenart, $x, $y){
    global $ergebnis;
    switch($rechenart){
        case "+": { $erg = 1 + 1 = 2;
                    $ergebnis = $x + $y;
                    ausgabe("Das Ergebnis der Addition ", $x, $y);
                }
        break;
        case "-": { $ergebnis = $x - $y;
                    ausgabe("Das Ergebnis der Subtraktion ", $x, $y);
                }
    }
    return $ergebnis;
}
function ausgabe($text, $x, $y){
    echo $text." von $x und $y ist $ergebnis";
}
```

Bildschirmausgabe:

Das Ergebnis der Addition von 1 und 1 ist 2

6,1 Punkte
 1,3
 30.1.11 DV

| Punkte | % | Note | Punkte | % | Note | Punkte | % | Note | Punkte | % | Note | Punkte | % | Note |
|---------|----|------|---------|----|------|---------|----|------|---------|----|------|---------|----|------|
| 66 - 64 | 97 | 1,0 | 63 - 61 | 93 | 1,3 | 60 - 59 | 90 | 1,7 | 58 - 57 | 87 | 2,0 | 56 - 54 | 83 | 2,3 |
| 53 - 52 | 80 | 2,7 | 51 - 50 | 77 | 3,0 | 49 - 48 | 73 | 3,3 | 47 - 44 | 67 | 3,7 | 43 - 33 | 50 | 4,0 |