

## Übungsaufgaben Assembler 2 :

- 1) Programmieren Sie eine Funktion, welche beim Aufruf immer den Inhalt von R5 tauscht. In R5 soll abwechselnd 0x10 und 0x20 stehen.
- 2) Programmieren Sie eine Funktion/Subroutine, die in einer Schleife ein Register immer um den Wert 2 erhöht, so lange bis das Register einen Wert **größer gleich** 80h hat.
- 3) Programmieren Sie eine Funktion, die den Wert in R5 mit dem Faktor 8 multipliziert. Die Funktion soll weiterhin prüfen, ob ein Überlauf auftritt und in diesem Fall den Wert in R8 auf 50h setzen.
- 4) Programmieren Sie eine Funktion, die einen 32-bit Wert (R5 und R6) mit dem Faktor 4 multipliziert und den Überlauf überwacht. Bei einem Überlauf soll R7 zu 1 gesetzt werden.
- 5) Programmieren Sie eine Funktion, die den Wert der Register R5, R6 und R7 addiert. Das Ergebnis soll in R8 und R9 stehen.
- 6) Programmieren Sie eine Funktion, welche den Wert der Register R8 und R9 vertauscht. Benutzen Sie hier kein weiteres Register zum Zwischenspeichern, sondern verwenden Sie den Stack!
- 7) Programmieren Sie eine Funktion, welche bei jedem Aufruf das Bit 1 des Registers R5 invertiert (= „toggelt“).
- 8) Programmieren Sie eine Funktion die prüft, ob der Wert in R5 kleiner als 0x80 und größer als 0x20 ist. Ist dies der Fall, so soll R7 auf 0x0001 gesetzt werden. Ist dies nicht der Fall, so soll R7 zu 0x0000 gesetzt werden.
- 9) Programmieren Sie eine Funktion, die bei jedem Aufruf das Register R5 um den Wert des Registers R6 erhöht. Das Register R6 soll einen Startwert von 0x0000 haben und bei jedem Funktionsaufruf selbst um +1 erhöht werden.
- 10) Programmieren Sie in einer Endlosschleife ein „Lauflicht“ mit Register R5. Der Startwert von R5 soll 0x0001 sein. Die anfängliche „1“ ganz rechts soll nach links wandern bis zum 15. Bit. Dann soll die Richtung umkehren und die „1“ soll wieder nach rechts wandern bis zum 0. Bit. Es soll immer nur ein Bit in R5 logisch „1“ sein, alle anderen Bits sollen „0“ sein.