Georgios Markou MPT Test2

a)

1) Beispiel: PUSH {R2,LR}
Der Wert von R15 (PC) wird auf den Stack gepusht. Anschließend wird er vom Stack wieder in das PC-Register gespeichert.

2) Beispiel: bl myFunc
Wir springen zur Adresse die in LR gespeichert war

Die Variante 1 bietet den Vorteil, dass Schachtelung implizit unterstützt wird, weil der PC vor dem Sprung auf dem Stack gespeichert wird.

b) 1) a: r24; b: r22
2) r2-r17, r28-r29 (r29:r28 = Y Pointer)

c) 1) a: r0; b: r1
2) r4-r11

d) 1) c
2) b
3) a
4) d

e) Synchron bedeutet, dass der Interrupt, immer bei den gleichen Bedingungen ausgelöst wird (vorhersehbar).
Er wird von der CPU selbst ausgelöst und ist für das laufende Programm bestimmt.
Wird auch als Exception bezeichnet.

Sprünge sind auch synchron, weil schon beim Kompilieren bekannt ist wo gesprungen wird, es ist immer vorhersehbar/bekannt.

Nach der Abarbeitung dieser Sprünge und SW-Interrupts wird das eigentliche Programm weiter ausgeführt.

f) Das Sprungziel ist bei SW-Interrupts nicht durch die Instruktion beliebig anpassbar. Es wird an eine sichere Adresse gesprungen, diese ist durch das System in der Vektortabelle vorgegeben.

In Assembler-Code wird an die Adresse die PC enthält gesprungen. In unserem Fall (beim ARM Cortex M-3) ist PC das Register R15.

g) höchste Priorität (absteigend): RESET, NMI (Non Maskable Interrupt), HardFault

Reset hat die Priorität -3. NMI hat die Priorität -2. HardFault hat die Priorität -1.
Alle drei sind nicht konfigurierbar, haben negative Prioritäten und sind nicht synchron.

Alle anderen Vektoren haben programmierbare Priorität. Priorität muss im Bereich von 0 bis einschließlich 255 sein(je kleiner die Zahl desto höher die Priorität).