From last time

What are the differences between these three ARM instructions: (3 marks)

ADD R1, R2, R3
ADDS R1, R2, R3 and ADDEQ R1, R2, R3

ADDS does the ADD and then does “CMP R1, #0”
ADDEQ only does the ADD if e.g. the previous CMP compared two equal values

Explain how using “S” (as for ADDS above) can sometimes help to make a code sequence shorter and faster, giving an example. (3 marks)
e.g. evaluating condition in an “if” or “while”

e.g. if (a+b == 0) {
...
}

LDR R0, a
LDR R1, b
ADD R0, R0, R1 }
ADDS R0, R0, R1
CMP R0,#0
BNE skip

...
Explain how using conditional instructions can sometimes help to make a code sequence shorter and faster, giving an example. (3 marks)

e.g. in code for an “if” or “while”

e.g. while (a == b) {a = a + b;}

LDR R0, a
LDR R1, b
start CMP R0, R1
BNE skip

ADD R0, R0, R1

ADD E Q R0, R0, R1
B start

...
Question: Cash-till program

What are the offsets for these LDR and STR instructions?

LDR R3, minimum
CMP R2, R3
BLT nodiscount
LDR R4, discount
SUB R5, R2, R4
STR R5, total

nодiscount SVC 2
total DEFW 1534
minimum DEFW 2000
discount DEFW 100

STR R5, total ; offset from PC = 2 - 2 = 0 words = 0 bytes
LDR R4, discount ; offset from PC = 6 - 2 = 4 words = 16 bytes
LDR R3, minimum ; offset from PC = 8 - 2 = 6 words = 24 bytes
Question: Optimise using Conditional Instructions

**e.g.** \( \text{ADD} \rightarrow \text{ADDEQ} \)

\[
\begin{align*}
\text{ADRL} & \quad R1, \text{message} \\
\text{loop} & \quad \text{LDRB} \quad R0, [R1], #1 \\
& \quad \text{CMP} \quad R0, #0 \\
& \quad \text{SVCNE} \quad 0 \\
& \quad \text{BNE} \quad \text{loop}
\end{align*}
\]