Recap

1. **Capacitor** - Do not react to instantaneous changes in voltage
clock cycle 16 MHz

1 cycle = 62.5 ns

Simulator vs Real Board

$\Delta t$ = number of cycles / given instruction

$\Delta t$ \downarrow \text{Pipelining}

ADD $R1, R2, R3$
Today

1. Subroutines C & ASM
2. Parameter passing
   - Call-by-Value
   - Call-by-Reference
3. Pointers in C
4. Arrays
Subroutines in ASM/C

Call: BL Sub

Var = Sub( )

Ret: BX LR

return( )

Implicit passing using registers

I/P = 0
ARM AAPCS

ASM

```
MOV  R0, #5
BL  Square
J  R0 has 25
```

\[ \text{uint16}_t y; \]
\[ \text{uint8}_t x; \]
\[ x = 5; \]
\[ y = \text{Square}\( x \); \]

Box LR

\[ \text{uint16}_t \text{ func}( \text{uint8}_t \text{ in1}, \text{uint16}_t \text{ in2}) \]
\[ \text{uint16}_t \text{ res}; \]
\[ \text{return}( \text{res}); \]

8 = func(51, 5158)
AAPCS Rules

① up to 4 inputs passed in R0-R3
② Single s/p returned in R0
③ R0-R3, R12 can be manipulated by a subroutine \( \Rightarrow \) caller cannot expect these to remain intact
④ R4-R11 can be used by subroutine but must restore them to original values on return

LIFO

SAVE  PUSH [R4-R11]

Done using Stack

Restore

PDP [R4-R11]
AAPCS Rules

1. Up to 4 inputs passed in R0-R3
2. Single s/p returned in R0
3. R0-R3, R12 can be manipulated by a subroutine => caller cannot expect these to remain intact
4. R4-R11 can be used by subroutine but must restore them to original values on return

Stack:
- Push [R4-R11]
- Save
- Restore
int main(void)
{
    int x = 3;
    int y;
    int z;

    PUSH {R4-R11, R12, R13, R14}
    
    L0: if (y > 5)
    {
        x = y;
        goto L0;
    }

    SP = 0x2000;
    variables
    
    RAM

    } lower seg number to lower memory address
Push $\{R_1, R_2\}$

$R_7 \rightarrow 17$

$R_6 \rightarrow 5$

$PDP \{R_7, R_6\}$

$PDP \{R_7\}$

$PDP \{R_6\}$

$SP \rightarrow 5$

$17$
extern App.c

void Delay(void);

int main()
{
    Delay();
}

ASM Delay.s

AREA __

EXPORT Delay

Delay

BX LR

ALIGN

END
Module (Device Driver)

Modoh (Interface)

void Init()

void InitRead()

void Write(int32_t)

Public functions that other modules can call

# include "Modoh"

\[
\begin{align*}
\text{Init} & \equiv \frac{x}{2} \\
\text{Read} & \\
\text{Write} & \\
\text{Reset} &
\end{align*}
\]

\[
\begin{align*}
a & = 5 \\
\text{Sub}(x) & \\
\text{Sub}(5) &
\end{align*}
\]