

CPE/EE 421/521

## The MSP430 - Low Power

Dr. Rhonda Kay Gaede

**UAH**

**UAH**

**CPE/EE 421/521**

### Power as a Design Constraint

---

**Power becomes a first class architectural design constraint**

- Why worry about power?
  - Battery life in portable and mobile platforms
  - Power consumption in desktops, server farms
    - Cooling costs, packaging costs, reliability, timing
    - Power density: 30 W/cm<sup>2</sup> in Alpha 21364 (3x of typical hot plate)
  - Environment?
    - IT consumes 10% of energy in the US

## Components of Power in CMOS Systems

---

Dynamic power consumption

↓

Power due to short-circuit current during transition

↓

Power due to leakage current

↓

$$P = ACV^2f + \tau AVI_{short}f + V I_{leak}$$

## Components of Power in CMOS Systems: Dynamic Power Consumption

---

**C** – Total capacitance seen by the gate's outputs  
Function of wire lengths, transistor sizes, ...

**V** – Supply voltage  
Trend: has been dropping with each successive fab

$$ACV^2f$$

**A** - Activity of gates  
How often on average do wires switch?

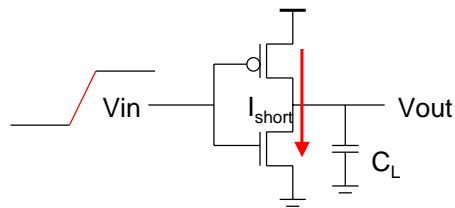
**f** – clock frequency  
Trend: increasing ...

**Reducing Dynamic Power**

- 1) Reducing V has quadratic effect; Limits?
- 2) Lower C - shrink structures, shorten wires
- 3) Reduce switching activity - Turn off unused parts or use design techniques to minimize number of transitions

## Components of Power in CMOS Systems: Short-Circuit Power Consumption

$$\tau AV I_{\text{short}} f$$



Finite slope of the input signal causes a direct current path between  $V_{DD}$  and GND for a short period of time during switching when both the NMOS and PMOS transistors are conducting

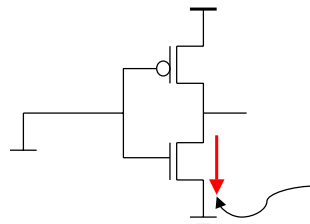
### Reducing Short-circuit

- 1) Lower the supply voltage  $V$
- 2) Slope engineering – match the rise/fall time of the input and output signals

Fall 2004 Page 5 of 24

## Components of Power in CMOS Systems: Leakage Power Consumption

$$VI_{\text{leak}}$$



Sub-threshold current

Sub-threshold current grows exponentially with increases in temperature and decreases in  $V_t$

Fall 2004 Page 6 of 24

## CMOS Power Equations

---

$$P = ACV^2f + \tau AVI_{\text{short}}f + V I_{\text{leak}}$$

Reduce the supply voltage,  $V$

$$f_{\text{max}} \propto \frac{(V - V_t)^2}{V}$$

Reduce threshold  $V_t$

$$I_{\text{leak}} \propto \exp\left(-\frac{qV_t}{kT}\right)$$

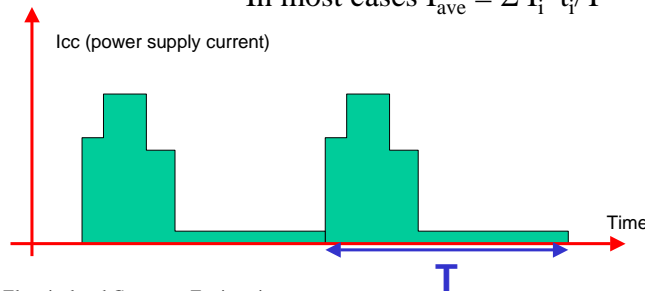
## Power Reduction Strategies

---

- **Dynamic power consumption**
  - charge/discharge of the capacitive load on each gate's output
  - frequency
- **Control activity**
  - reduce power supply voltage
  - reduce working frequency
  - turn off unused parts (module enables)
  - use low power modes
  - interrupt driven system
- **Minimize the number of transitions**
  - instruction formats, coding?

## Average Power Consumption

- Dynamic power supply current
  - Set of modules that are periodically active
  - Typical situation – real time cycle T
  - $I_{ave} = \int I_{cc}(t)dt / T$
  - In most cases  $I_{ave} = \sum I_i * t_i / T$



Fall 2004 Page 9 of 24

## Low-Power Concept: Basic Conditions for Burst Mode

The example of the heat cost allocator shows that the current of the non-activity period dominates the current consumption.

	Measure	Process data	Real-Time Clock	LCD Display
$I_{AVG} =$	$I_{Measure}$	$+ I_{Calculate}$	$+ I_{RTC}$	$+ I_{Display}$
	$= I_{ADC} * t_{Measure} / T$	$+ I_{active} * t_{calc} / T$	$+ I_{active} * t_{RTC} / T$	$+ I_{Display}$
	$= 3mA * 200\mu s / 60s$	$+ 0.5mA * 10ms / 60s$	$+ 0.5mA * 0.5ms / 60s$	$+ 2.1\mu A$
	$= 10nA$	$+ 83nA$	$+ 4nA$	$+ 2.1\mu A$
$I_{AVG} \equiv$				<b>2.1μA</b>

The **sleep current dominates** the current consumption!

The currents are related to the sensor and  $\mu C$  system. Additional current consumption of other system parts should be added for the total system current

Fall 2004 Page 10 of 24

## Battery Life

---

- Battery Capacity BC – [mAh]
- Battery Life
  - $BL = BC / I_{ave}$
- In the previous example, standard 800 mAh batteries will allow battery life of:
  - $BL = 750 \text{ mAh} / 2.1 \mu\text{A} \approx 44 \text{ years} !!!$
- Conclusion:
  - Power efficient modes
  - Interrupt driven system with processor in idle mode

Fall 2004 Page 11 of 24

## Factors Influencing Power Consumption

---

- Logic
  - Clock tree (up to 30% of power)
  - Clock gating (turn off branches that are not used)
  - Half frequency clock (both edges)
  - Half swing clock (half of Vcc)
  - Asynchronous logic
    - completion signals
    - testing
- Architecture
  - Parallelism (increased area and wiring)
  - Speculation (branch prediction)
  - Memory systems
    - Memory access (dynamic)
    - Leakage
    - Memory banks (turn off unused)
  - Buses
    - 32-64 address/data, (15-20% of power)
    - Gray Code, Code compression

Fall 2004 Page 12 of 24

## Factors Influencing Power Consumption

---

- Operating System
  - Finish computation “when necessary”
  - Scale the voltage
    - Application driven
    - Automatic
- System Architecture
  - Power efficient and specialized processing cores
  - A “convergent” architecture
  - Trade-off
    - AMD K6 / 400MHz / 64KB cache – 12W
    - XScale with the same cache 450 mW @ 600 MHz (40mW@150MHz)
- Other issues
  - Leakage current – Thermal runaway
  - Voltage clustering (low Vthreshold for high speed paths)

## Operating Modes-General

---

The MSP430 family was developed for ultra-low power applications and uses different levels of operating modes. The MSP430 operating modes give advanced support to various requirements for ultra-low power and ultra-low energy consumption. This support is combined with an intelligent management of operations during the different module and CPU states. An interrupt event wakes the system from each of the various operating modes and the RETI instruction returns operation to the mode that was selected before the interrupt event.

The ultra-low power system design which uses complementary metal-oxide semiconductor (CMOS) technology, takes into account three different needs:

- The desire for speed and data throughput despite conflicting needs for ultra-low power
- Minimization of individual current consumption
- Limitation of the activity state to the minimum required by the use of low power modes

## Low-Power Mode Control

---

There are four bits that control the CPU and the main parts of the operation of the system clock generator:

- CPUOff,
- OscOff,
- SCG0, and
- SCG1.

These four bits support discontinuous active mode (AM) requests, to limit the time period of the full operating mode, and are located in the status register. The major advantage of including the operating mode bits in the status register is that the present state of the operating condition is saved onto the stack during an interrupt service request. As long as the stored status register information is not altered, the processor continues (after RETI) with the same operating mode as before the interrupt event.

## Low-Power Operating Modes and Interrupts

---

### Enter interrupt routine

The interrupt routine is entered and processed if an enabled interrupt awakens the MSP430:

- The SR and PC are stored on the stack, with the content present at the interrupt event.
- Subsequently, the operation mode control bits OscOff, SCG1, and CPUOff are cleared automatically in the status register.

### Return from interrupt

Two different modes are available to return from the interrupt service routine and continue the flow of operation:

- Return with low-power mode bits set. When returning from the interrupt, the program counter points to the next instruction. The instruction pointed to is not executed, since the restored low power mode stops CPU activity.
- Return with low-power mode bits reset. When returning from the interrupt, the program continues at the address following the instruction that set the OscOff or CPUOff-bit in the status register. To use this mode, the interrupt service routine must reset the OscOff, CPUOff, SCGO, and SCG1 bits on the stack. Then, when the SR contents are popped from the stack upon RETI, the operating mode will be active mode (AM).

## Software Configuration of Low-Power Operating Modes

---

**There are six operating modes that the software can configure:**

- Active mode AM; SCG1=0, SCG0=0, OscOff=0, CPUOff=0: CPU clocks are active
- Low power mode 0 (LPM0); SCG1=0, SCG0=0, OscOff=0, CPUOff=1:
  - CPU and MCLK are disabled
  - SMCLK and ACLK remain active
- Low power mode 1 (LPM1); SCG1=0, SCG0=1, OscOff=0, CPUOff=1:
  - CPU and MCLK are disabled
  - DCO's dc generator is disabled if the DCO is not used for MCLK or SMCLK when in active mode. Otherwise, it remains enabled.
  - SMCLK and ACLK remain active
- Low power mode 2 (LPM2); SCG1=1, SCG0=0, OscOff=0, CPUOff=1:
  - CPU, MCLK, and SMCLK are disabled
  - DCO oscillator automatically disabled because it is not needed for MCLK or SMCLK, DCO's dc-generator remains enabled
  - ACLK remains active

Fall 2004 Page 17 of 24

## Software Configuration of Low-Power Operating Modes (continued)

---

- Low power mode 3 (LPM3); SCG1=1, SCG0=1, OscOff=0, CPUOff=1:
  - CPU, MCLK, and SMCLK are disabled
  - DCO oscillator is disabled
  - DCO's dc-generator is disabled
  - ACLK remains active
- Low power mode 4 (LPM4); SCG1=X, SCG0=X, OscOff=1, CPUOff=1:
  - CPU, ACLK, MCLK, and SMCLK are disabled
  - DCO oscillator is disabled
  - DCO's dc-generator is disabled
  - Crystal oscillator is stopped

Fall 2004 Page 18 of 24

## Low-Power Operating Mode Details

---

### ❑ Low-Power Mode 0 and 1 (LPM0 and LPM1)

Low power mode 0 or 1 is selected if bit CPUOff in the status register is set. Immediately after the bit is set the CPU stops operation, and the normal operation of the system core stops. The operation of the CPU halts and all internal bus activities stop until an interrupt request or reset occurs. The system clock generator continues operation, and the clock signals MCLK, SMCLK, and ACLK stay active depending on the state of the other three status register bits, SCG0, SCG1, and OscOff.

The peripherals are enabled or disabled with their individual control register settings, and with the module enable registers in the SFRs. All I/O port pins and RAM/registers are unchanged. Wake up is possible through all enabled interrupts.

## Low-Power Operating Mode Details

---

### ❑ Low-Power Modes 2 and 3 (LPM2 and LPM3)

Low-power mode 2 or 3 is selected if bits CPUOff and SCG1 in the status register are set. Immediately after the bits are set, CPU, MCLK, and SMCLK operations halt and all internal bus activities stop until an interrupt request or reset occurs.

Peripherals that operate with the MCLK or SMCLK signal are inactive because the clock signals are inactive. Peripherals that operate with the ACLK signal are active or inactive according with the individual control registers and the module enable bits in the SFRs. All I/O port pins and the RAM/registers are unchanged. Wake up is possible by enabled interrupts coming from active peripherals or RST/NMI.

## Low-Power Operating Mode Details

---

### ❑ Low-Power Mode 4 (LPM4)

In low power mode 4 all activities cease; only the RAM contents, I/O ports, and registers are maintained. Wake up is only possible by enabled external interrupts.

Before activating LPM4, the software should consider the system conditions during the low power mode period . The two most important conditions are environmental (that is, temperature effect on the DCO), and the clocked operation conditions.

The environment defines whether the value of the frequency integrator should be held or corrected. A correction should be made when ambient conditions are anticipated to change drastically enough to increase or decrease the system frequency while the device is in LPM4.

Fall 2004 Page 21 of 24

## Assembly Support for Low-Power Modes

---

### ❑ The following example describes entering into low-power mode 0.

```

;==Main program flow with switch to CPUOff Mode=====
BIS #18h,SR ;Enter LPM0 + enable general interrupt GIE (CPUOff=1, GIE=1).
            ;The PC is incremented during execution of this instruction
            ;and points to the consecutive program step.
.....    ;The program continues here if the CPUOff bit is reset during
            ;the interrupt service routine. Otherwise, the PC retains its
            ;value and the processor returns to LPM0.

```

### ❑ The following example describes clearing low-power mode 0.

```

;==Interrupt service routine=====
.....    ;CPU is active while handling interrupts
BIC #10h,0(SP) ;Clears the CPUOff bit in the SR contents that were
            ;stored on the stack.
RETI      ;RETI restores the CPU to the active state because the
            ;SR values that are stored on the stack were manipulated.
            ;This occurs because the SR is pushed onto the stack
            ;upon an interrupt, then restored from the
            ;stack after the RETI instruction.

```

Fall 2004 Page 22 of 24

## C Support for Operating Modes

```

/* Low Power Modes coded with Bits 4-7 in SR*/ /* STATUS REGISTER BITS */
/* Begin #defines for assembler */

#ifndef __IAR_SYSTEMS_ICC
#define LPM0 CPUOFF
#define LPM1 SCG0+CPUOFF
#define LPM2 SCG1+CPUOFF
#define LPM3 SCG1+SCG0+CPUOFF
#define LPM4 SCG1+SCG0+OSCOFF+CPUOFF
/* End #defines for assembler */

#else /* Begin #defines for C */
#define LPM0_bits CPUOFF
#define LPM1_bits SCG0+CPUOFF
#define LPM2_bits SCG1+CPUOFF
#define LPM3_bits SCG1+SCG0+CPUOFF
#define LPM4_bits SCG1+SCG0+OSCOFF+CPUOFF

```

```

#define C 0x0001
#define Z 0x0002
#define N 0x0004
#define V 0x0100
#define GIE 0x0008
#define CPUOFF 0x0010
#define OSCOFF 0x0020
#define SCG0 0x0040
#define SCG1 0x0080

```



Fall 2004 Page 23 of 24

## C Support for Low-Power Operating Modes

□ ...

```
#include "In430.h"
```

```

#define LPM0 _BIS_SR(LPM0_bits) /* Enter LP Mode 0 */
#define LPM0_EXIT _BIC_SR(LPM0_bits) /* Exit LP Mode 0 */
#define LPM1 _BIS_SR(LPM1_bits) /* Enter LP Mode 1 */
#define LPM1_EXIT _BIC_SR(LPM1_bits) /* Exit LP Mode 1 */
#define LPM2 _BIS_SR(LPM2_bits) /* Enter LP Mode 2 */
#define LPM2_EXIT _BIC_SR(LPM2_bits) /* Exit LP Mode 2 */
#define LPM3 _BIS_SR(LPM3_bits) /* Enter LP Mode 3 */
#define LPM3_EXIT _BIC_SR(LPM3_bits) /* Exit LP Mode 3 */
#define LPM4 _BIS_SR(LPM4_bits) /* Enter LP Mode 4 */
#define LPM4_EXIT _BIC_SR(LPM4_bits) /* Exit LP Mode 4 */
#endif /* End #defines for C */

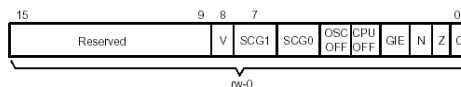
```

```
/* - in430.h -Intrinsic functions for the MSP430 */
```

```

unsigned short _BIS_SR(unsigned short);
unsigned short _BIC_SR(unsigned short);

```



Fall 2004 Page 24 of 24