HELLO!

To everyone on campus and off campus ☺

Graduate of AUK ☺

Who I am

Introduce yourself Please ☺

And what do you love to do the most

Let's discuss the syllabus

CPEG330L - Microprocessors & Interfacing Lab.

A few Key points to keep in mind

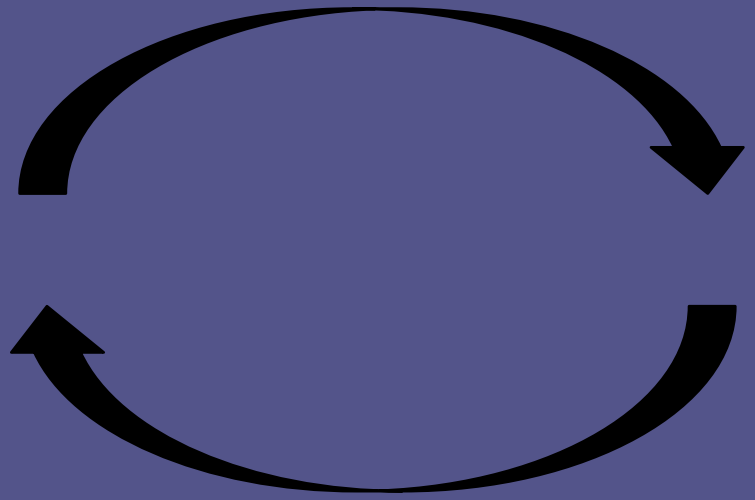
My office Hours:

Sunday 2:30 pm to 6:00 pm

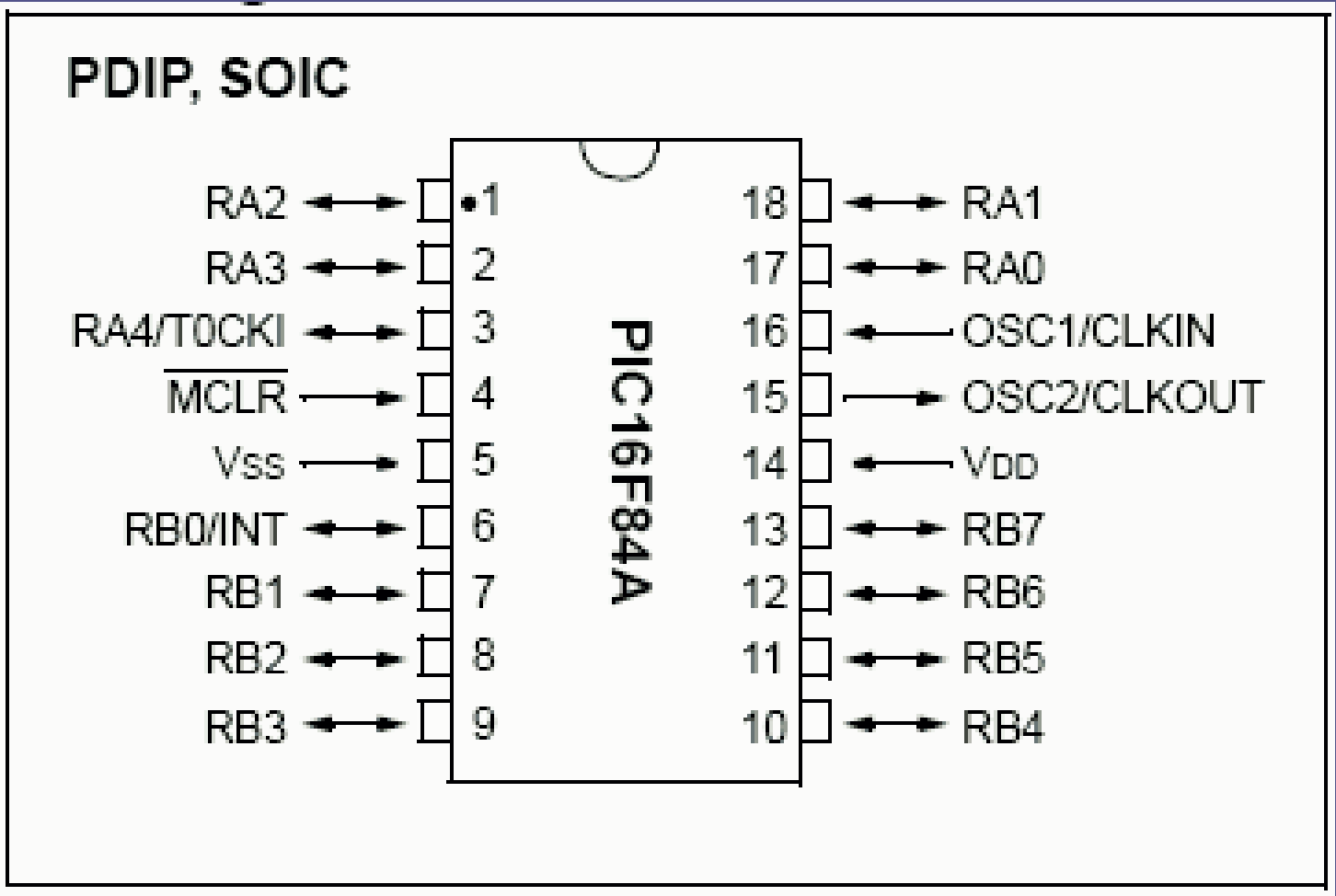
Monday 11:30 pm to 2:00pm

Course Schedule will be changing.

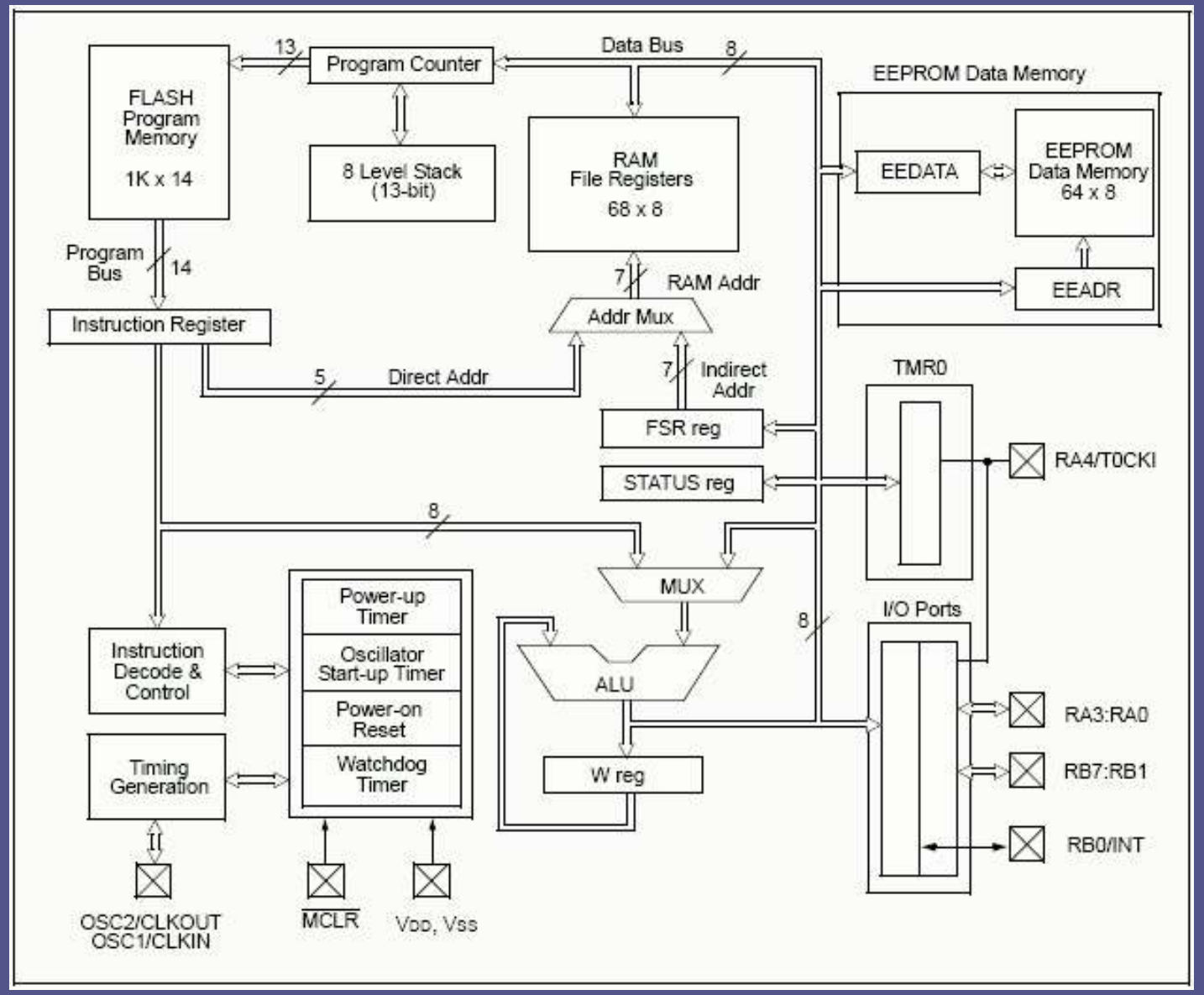
Reports will be submitted on Turnitin. Any reports showing large amounts of consecutively non-original work, or a similarity percentage over 50%, risks a grade of zero for the coursework portion of that lab.



	EVEN	Odd
Week 1	Lab [1] Exercise A Exercise B	Lab [1] Exercise C Exercise D
Week 2	Lab [1] Exercise C Exercise D	Lab [1] Exercise A Exercise B



The PIC16F84A



The PIC16F84A architecture

R/W-0	R/W-0	R/W-0	R-1	R-1	R/W-x	R/W-x	R/W-x
IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C
bit 7							bit 0

bit 7-6 **Unimplemented:** Maintain as '0'

bit 5 **RP0**: Register Bank Select bits (used for direct addressing)
01 = Bank 1 (80h - FFh)
00 = Bank 0 (00h - 7Fh)

bit 4 $\overline{TO}$: Time-out bit
1 = After power-up, CLRWDT instruction, or SLEEP instruction
0 = A WDT time-out occurred

bit 3 $\overline{PD}$: Power-down bit
1 = After power-up or by the CLRWDT instruction
0 = By execution of the SLEEP instruction

bit 2 **Z**: Zero bit
1 = The result of an arithmetic or logic operation is zero
0 = The result of an arithmetic or logic operation is not zero

bit 1 **DC**: Digit Carry/$\overline{borrow}$ bit (ADDWF, ADDLW, SUBLW, SUBWF instructions) (for $\overline{borrow}$, the polarity is reversed)
1 = A carry-out from the 4th low order bit of the result occurred
0 = No carry-out from the 4th low order bit of the result

bit 0 **C**: Carry/$\overline{borrow}$ bit (ADDWF, ADDLW, SUBLW, SUBWF instructions) (for $\overline{borrow}$, the polarity is reversed)
1 = A carry-out from the Most Significant Bit of the result occurred
0 = No carry-out from the Most Significant Bit of the result occurred

Note: A subtraction is executed by adding the twos complement of the second operand. For rotate (RRF, RLF) instructions, this bit is loaded with either the high or low order bit of the source register.

The status register

Let’s get started