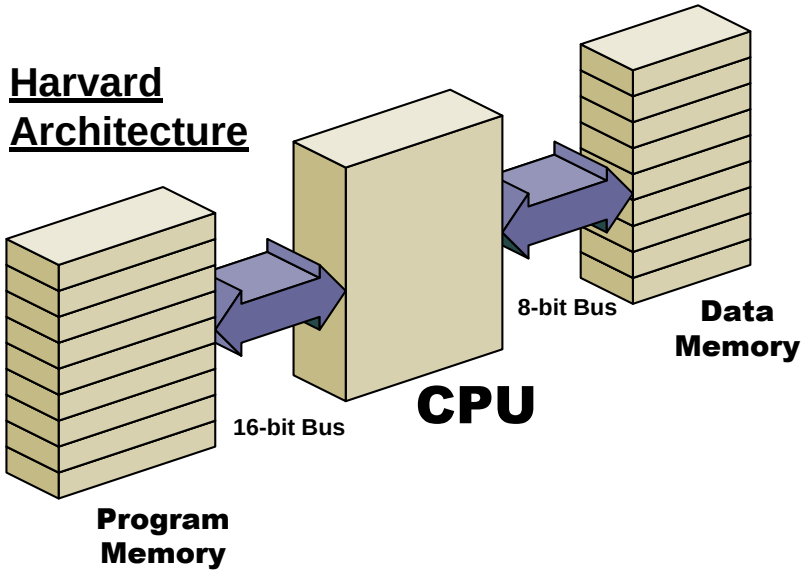
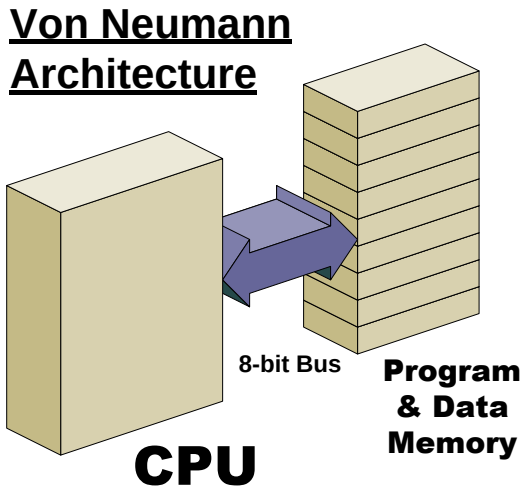
Microcontroller Architecture (PIC18F)

Processor Architecture Review

- Harvard
  - Separate data and program memory space (busses)
- Von-Neumann
  - Only one bus between CPU and memory



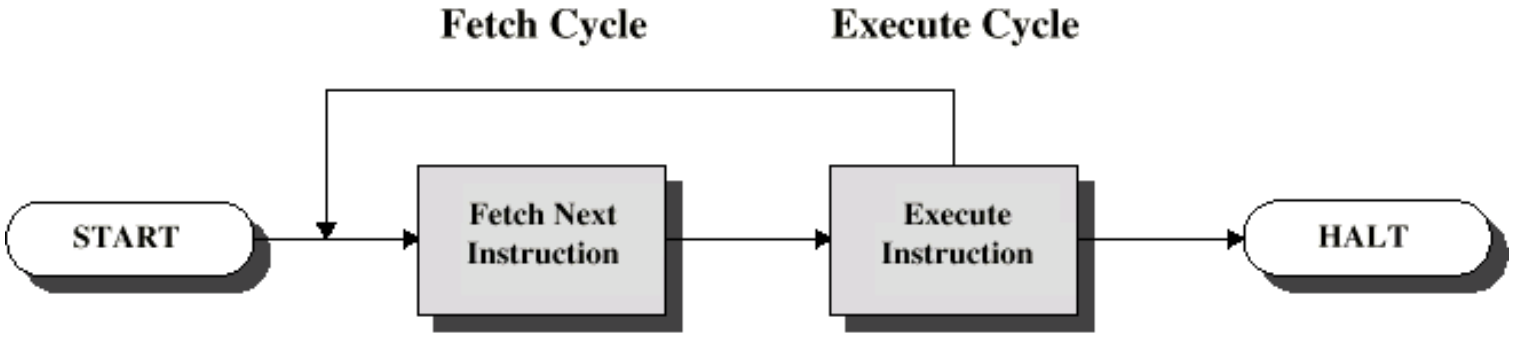


Processor Architecture Review

- RISC
  - A minimal set of simple instructions when combined can accomplish every needed operation
- CISC
  - A large set of complex instructions can singularly provide all needed operations

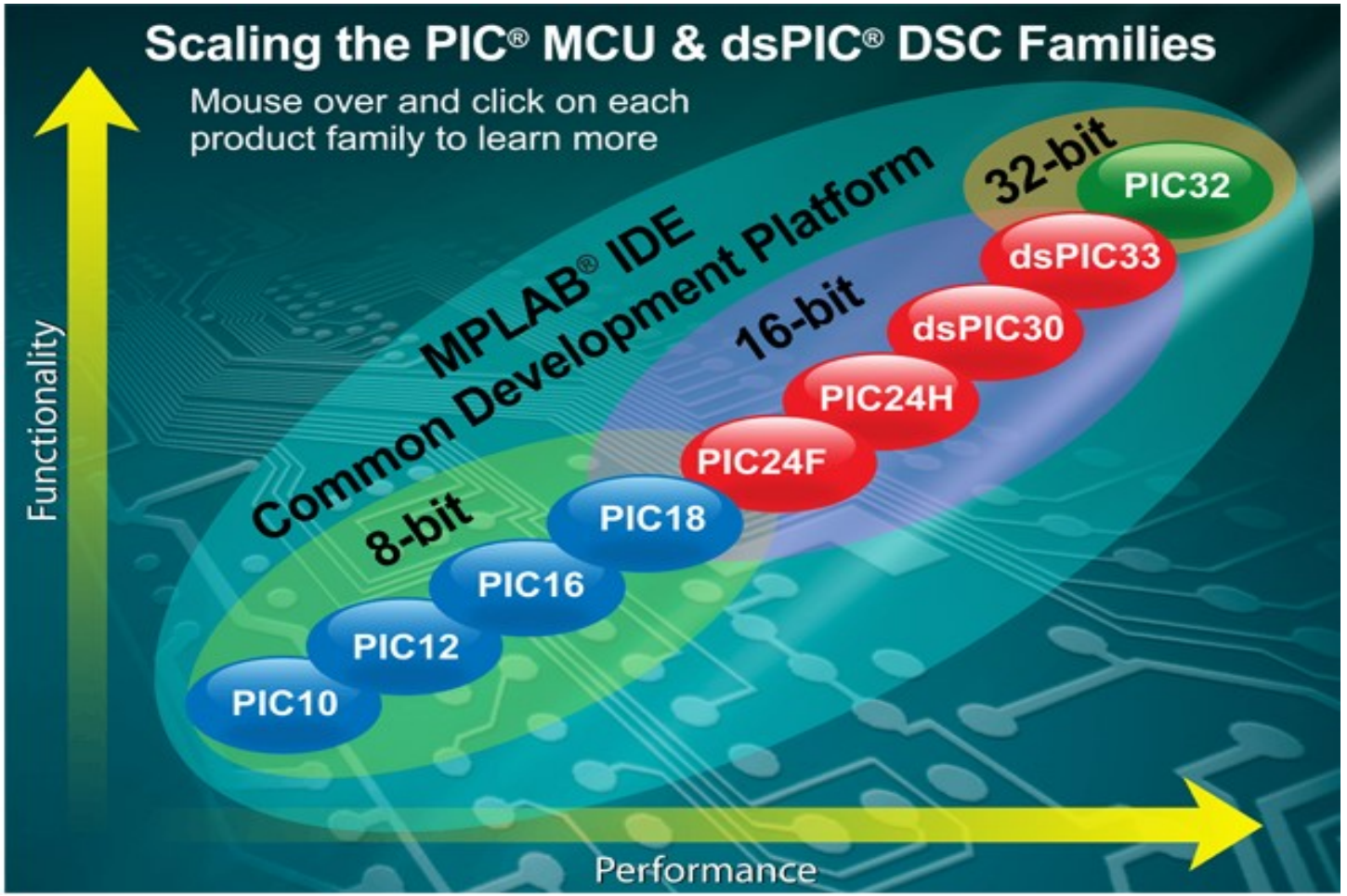
Processor Architecture Review

- Instruction Cycle

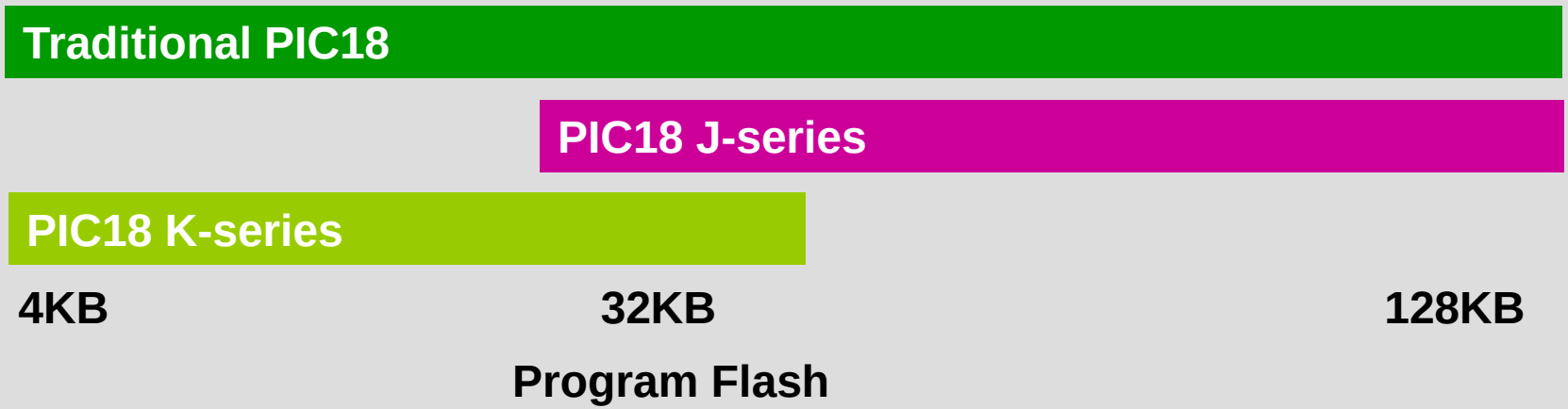


- Registers
  - Special purpose (PC)
  - General purpose

Microchip's MCU Families



PIC18 Families



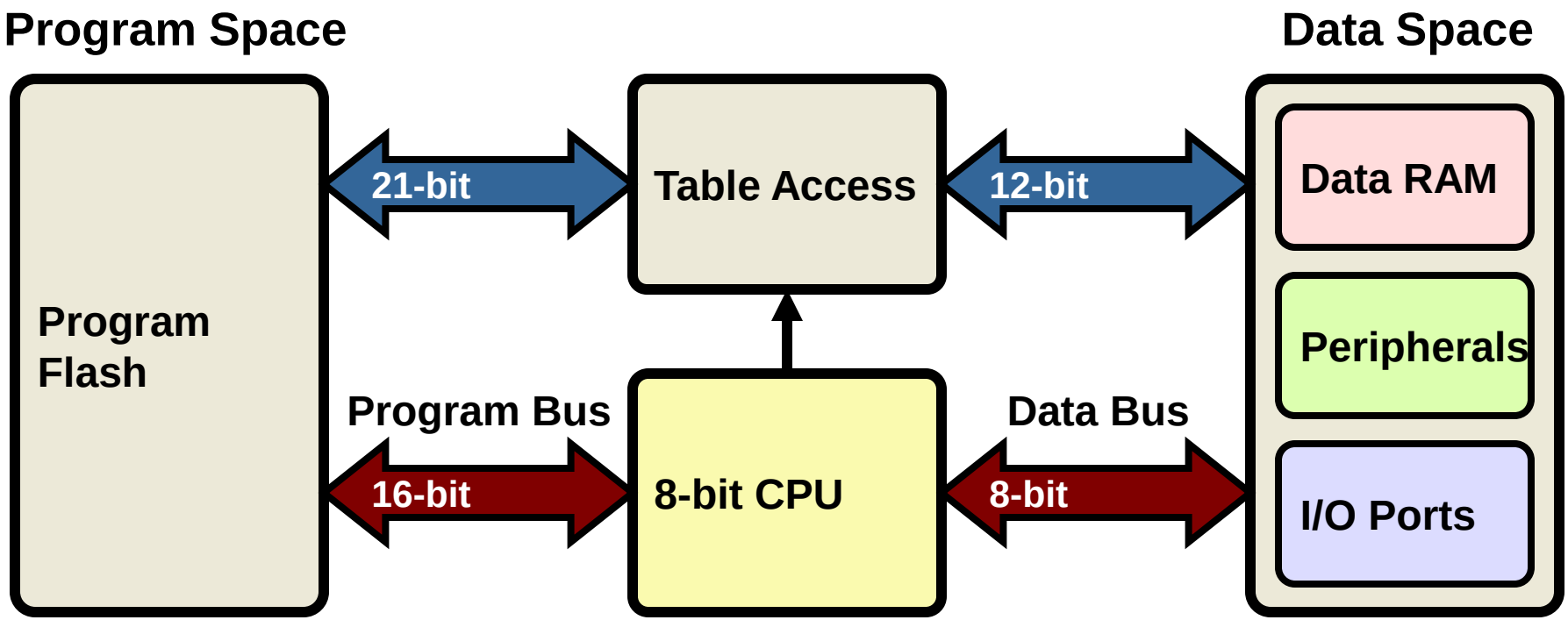
Typically products with higher memory also have higher pin-counts and higher levels of integrated peripherals

Traditional PIC18
40 MHz, 10 MIPS, 5V
Flash endurance 100k
EEPROM
Premium Features

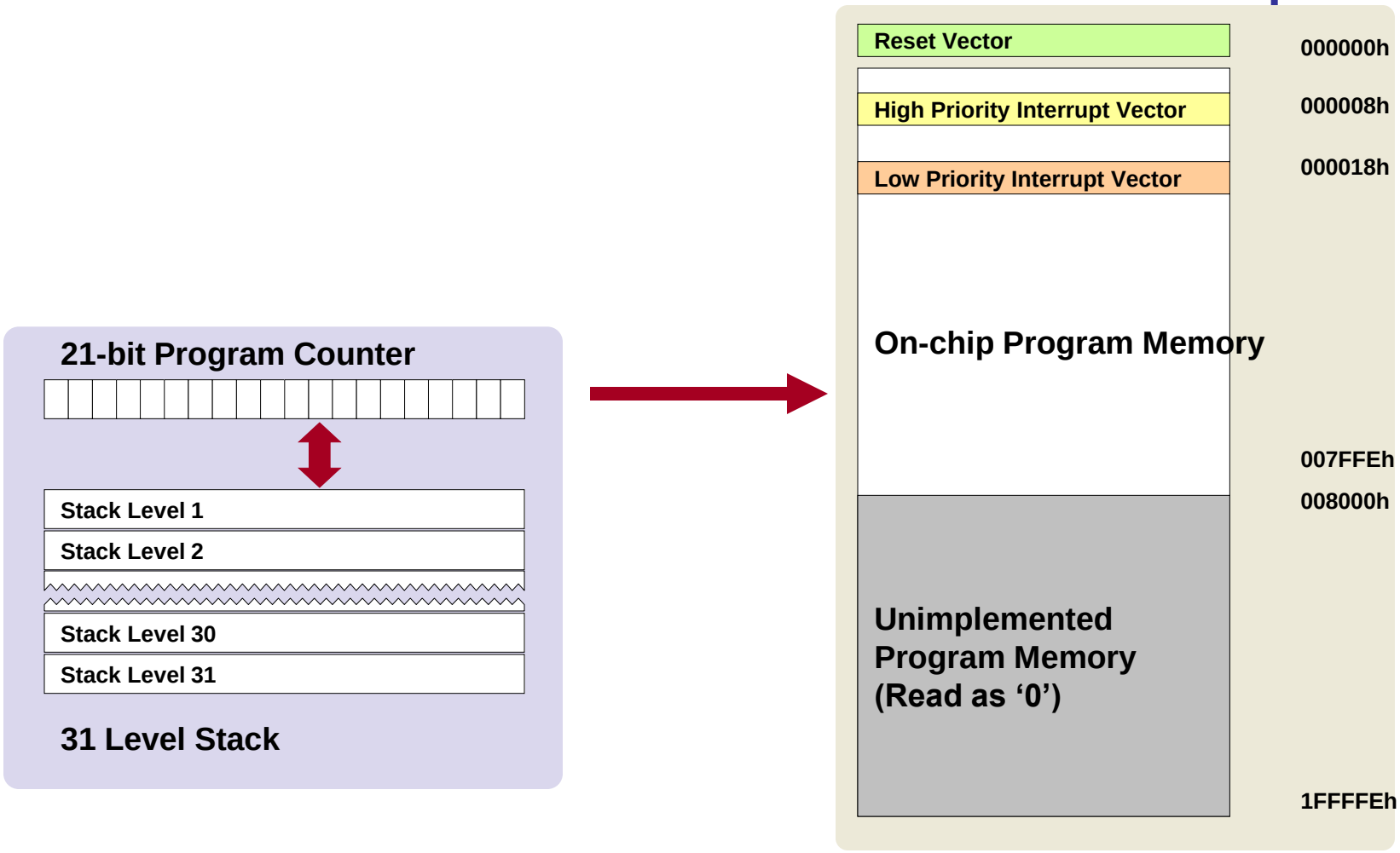
PIC18 J-series
40-48 MHz, 10-12 MIPS, 3V
Flash endurance 1k – 10k
Emulate EEPROM
Most cost effective >32KB Flash

PIC18 K-series
64MHz, 16 MIPS, 3V
Flash endurance 10k
EEPROM
Most cost effective <32KB Flash

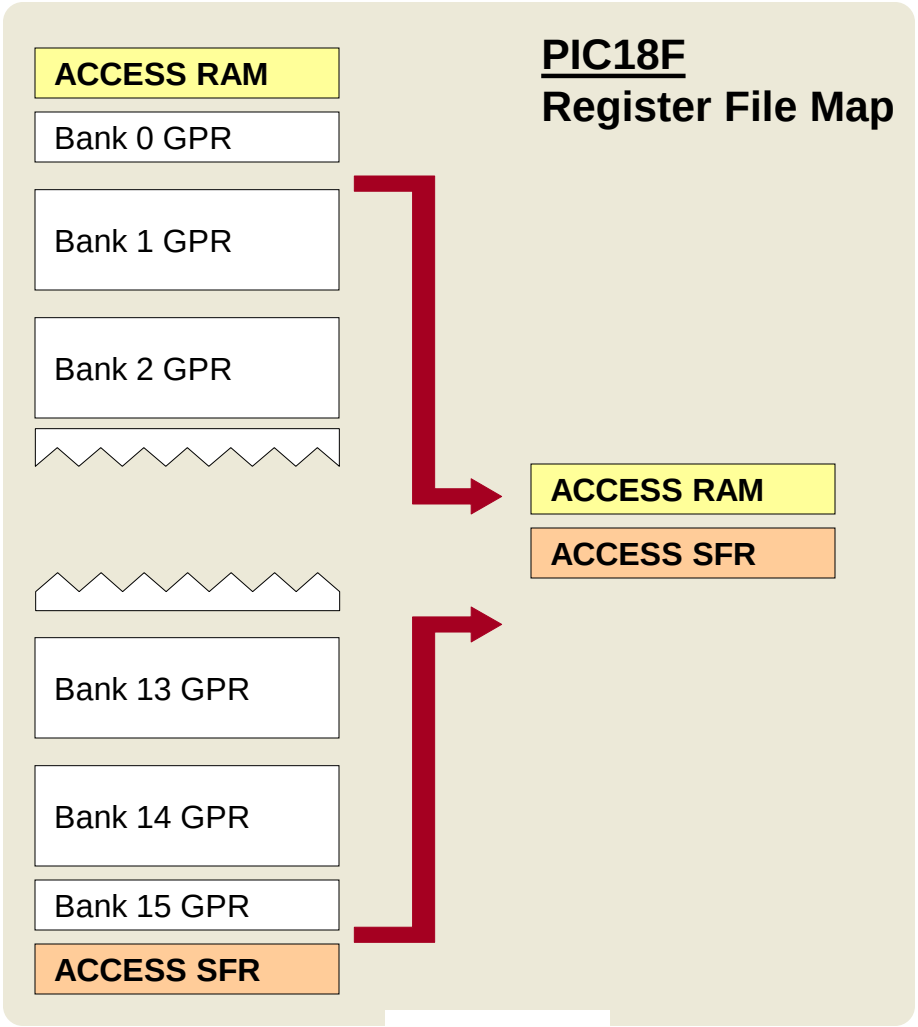
8-bit PIC® Architecture



PIC18 Program Memory Map



PIC18 Data Memory Map



PIC18 Registers

address	Name	Description
0xFFF	TOSU	Top of stack (upper)
0xFFE	TOSH	Top of stack (high)
0xFFD	TOSL	Top of stack (low)
0xFFC	STKPTR	Stack pointer
0xFFB	PCLATU	Upper program counter latch
0xFFA	PCLATH	High program counter latch
0xFF9	PCL	Program counter low byte
0xFF8	TBLPTRU	Table pointer upper byte
0xFF7	TBLPTRH	Table pointer high byte
0xFF6	TBLPTRL	Table pointer low byte
0xFF5	TABLAT	Table latch
0xFF4	PRODH	High product register
0xFF3	PRODL	Low product register
0xFF2	INTCON	Interrupt control register
0xFF1	INTCON2	Interrupt control register 2
0xFF0	INTCON3	Interrupt control register 3
0xFEF	INDF0 (1)	Indirect file register pointer 0
0xFEE	POSTINC0 (1)	Post increment pointer 0 (to GPRs)
0xFED	POSTDEC0 (1)	Post decrement pointer 0 (to GPRs)
0xFEC	PREINC0 (1)	Pre increment pointer 0 (to GPRs)
0xFEB	PLUSW0 (1)	Add WREG to FSR0

address	Name	Description
0xFEA	FSR0H	File select register 0 high byte
0xFE9	FSR0L	File select register 0 low byte
0xFE8	WREG	Working register
0xFE7	INDF1 (1)	Indirect file register pointer 1
0xFE6	POSTINC1 (1)	Post increment pointer 1 (to GPRs)
0xFE5	POSTDEC1 (1)	Post decrement pointer 1 (to GPRs)
0xFE4	PREINC1 (1)	Pre increment pointer 1 (to GPRs)
0xFE3	PLUSW1 (1)	Add WREG to FSR1
0xFE2	FSR1H	File select register 1 high byte
0xFE1	FSR1L	File select register 1 low byte
0xFE0	BSR	Bank select register
0xFDF	INDF2 (1)	Indirect file register pointer 2
0xFDE	POSTINC2 (1)	Post increment pointer 2 (to GPRs)
0xFDD	POSRDEC2 (1)	Post decrement pointer 2 (to GPRs)
0xFDC	PREINC2 (1)	Pre increment pointer 2 (to GPRs)
0xFDB	PLUSW2 (1)	Add WREG to FSR2
0xFDA	FSR2H	File select register 2 high byte
0xFD9	FSR2L	File select register 2 low byte
0xFD8	STATUS	Status register

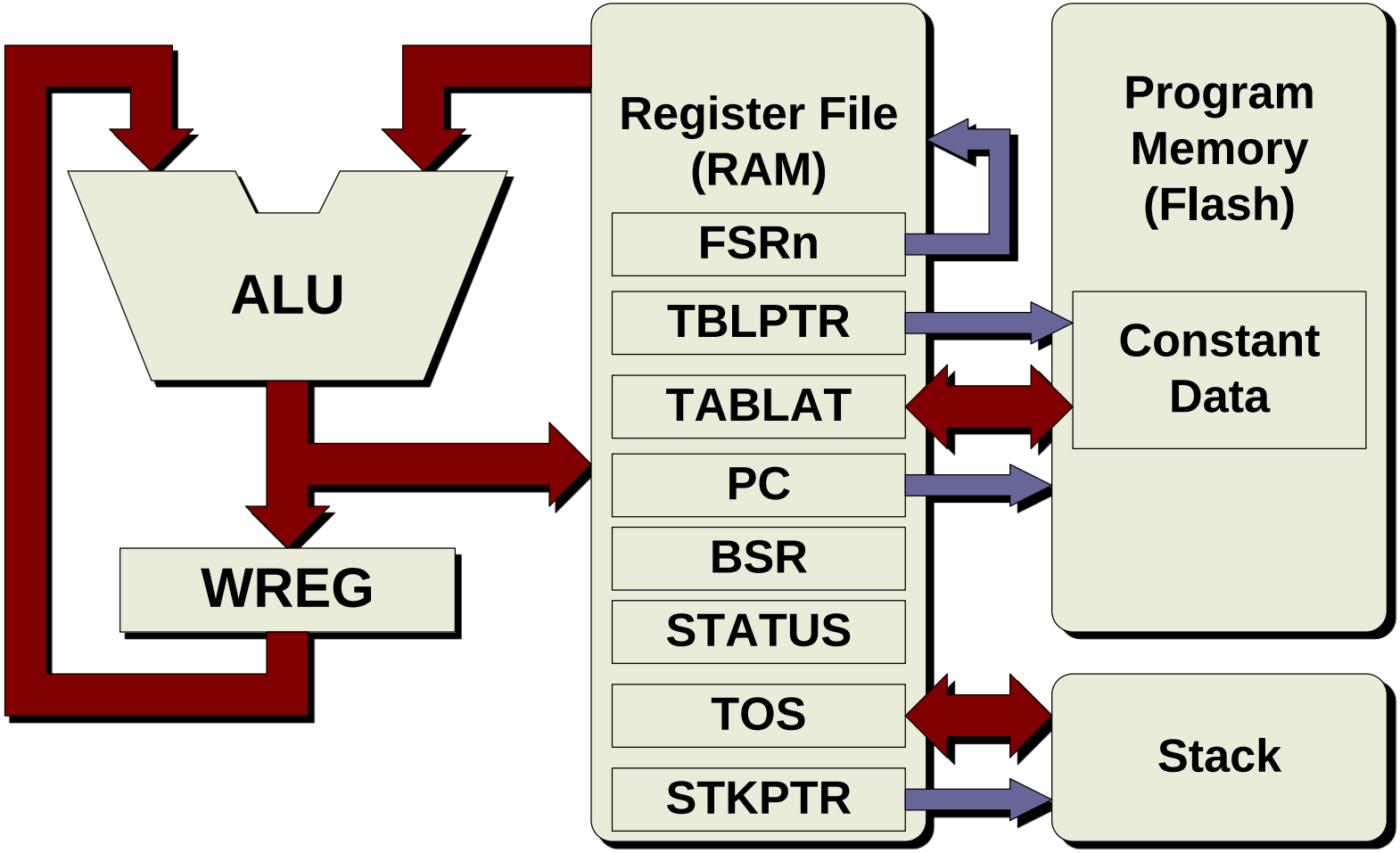
Note 1. This is not a physical register

PIC18 Registers

- Status register
  - Contains arithmetic status of the ALU
  - Bits set or cleared according to the device logic

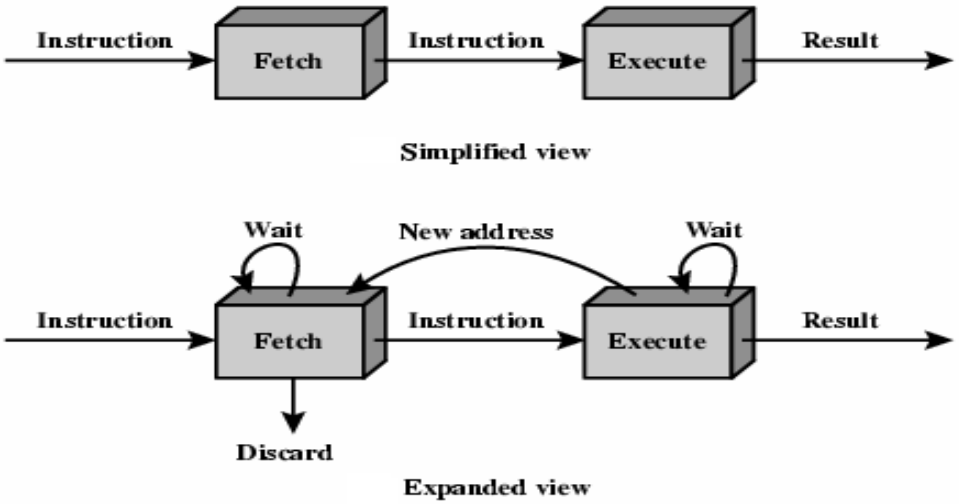
7	6	5	4	3	2	1	0
--	--	--	N	OV	Z	DC	C

PIC18 Programmer's Model



PIC18 Pipelining

- Allows processor to overlap the execution of several instruction to achieve higher instruction throughput
  - Utilizes the fact that different processor components are not fully utilized during the instruction execution process
  - Prefetches instructions during execution of current instructions

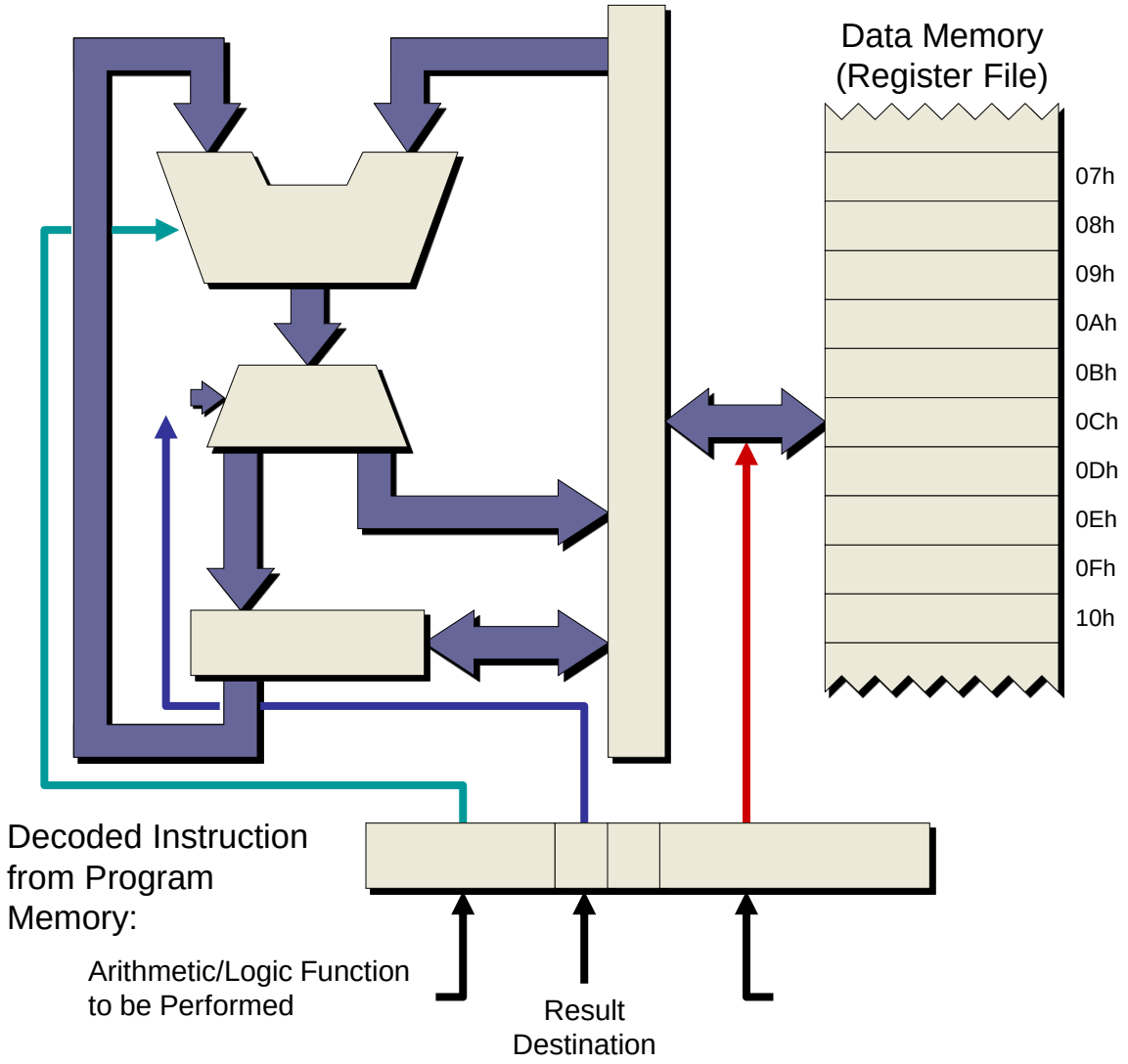


PIC18 Pipelining

- PIC18 utilizes a two-stage pipeline for instruction fetch and instruction execution

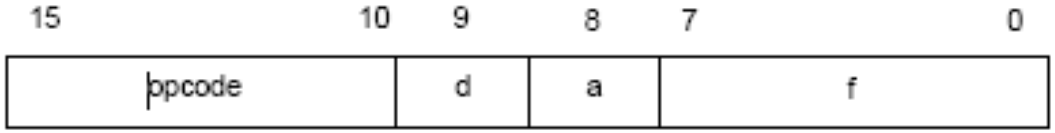
- Problems can arise from pipelining (Hazards)
  - Data dependency hazard
  - Control hazard
  - *Beyond scope of this class*

PIC18 Instruction Format



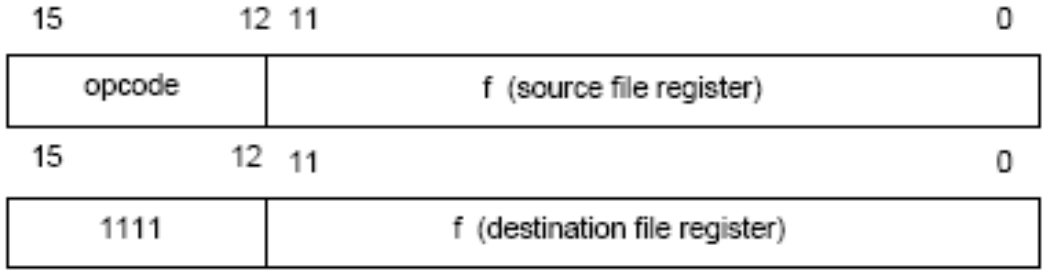
PIC18 Instruction Format

- Byte oriented file register instructions



d = 0 for result destination to be WREG register.
d = 1 for result destination to be file register (f)
a = 0 to force Access Bank
a = 1 for BSR to select bank
f = 8-bit file register address

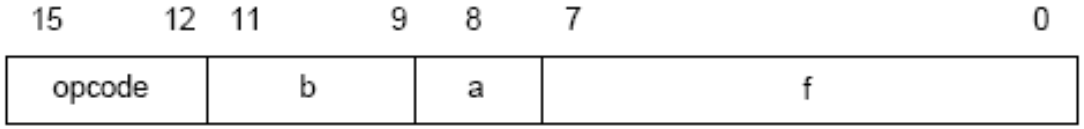
- Byte-to-byte move operations (2 words)



f = 12-bit file register address

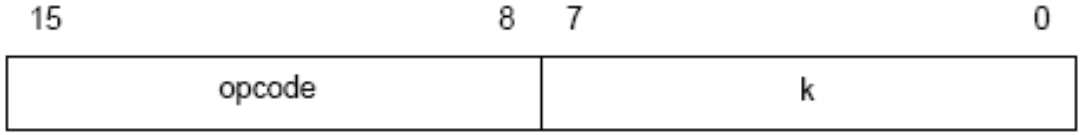
PIC18 Instruction Format

- Bit-oriented file register operations



b = 3-bit position of bit in the file register (f).
a = 0 to force Access Bank
a = 1 for BSR to select bank
f = 8-bit file register address

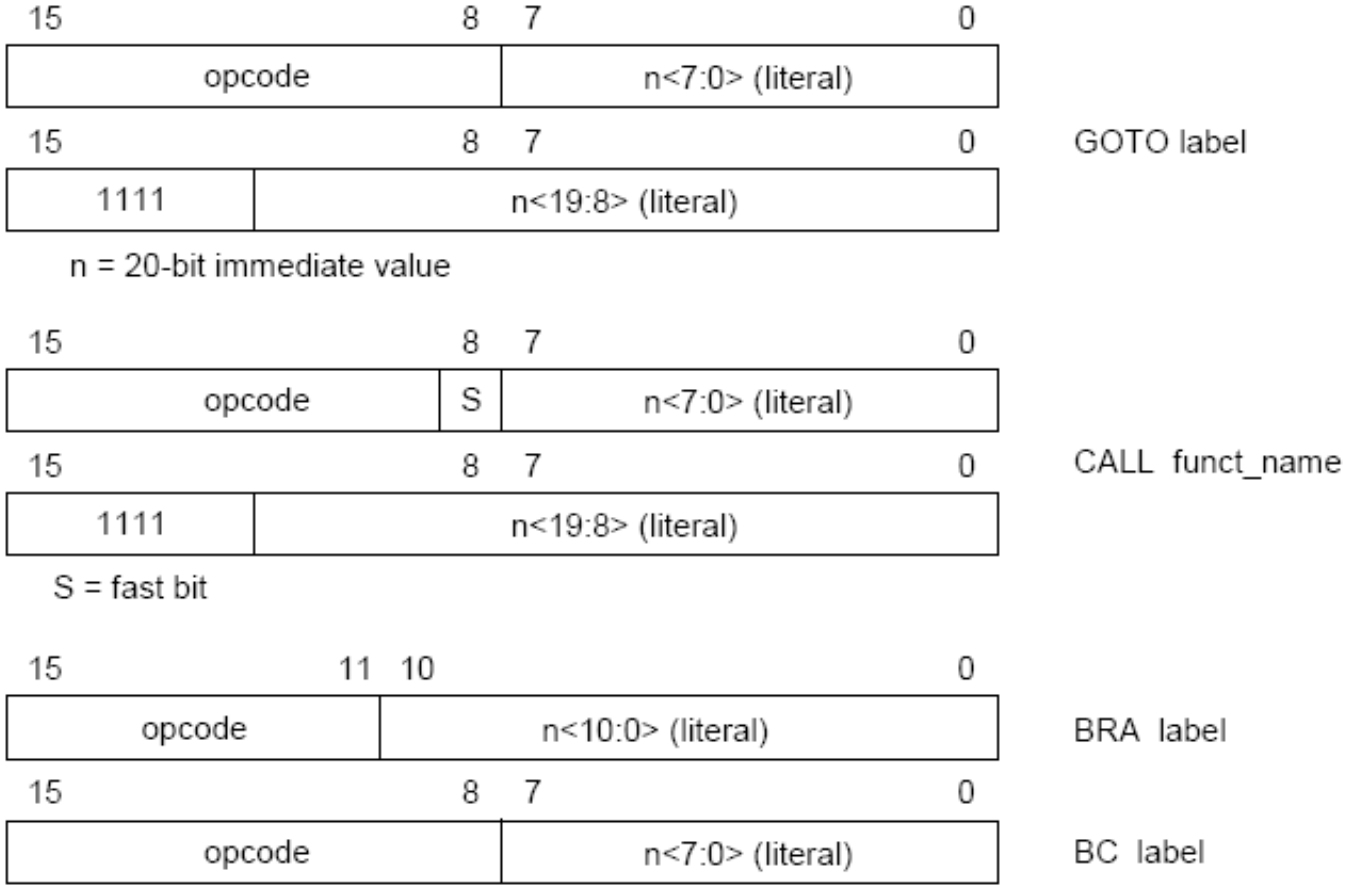
- Literal operations



k = 8-bit immediate value

PIC18 Instruction Format

- Control operations



PIC18 Instruction Format

- Mnemonic notation
  - ‘F’ or ‘W’ indicate that the source or destination address is the original register file location (F) or the working register (W)

- Number formats
  - Hex 0x7f, 20 (default), H’7f’ (alternative)
  - Binary B’10011100’
  - Decimal D’32’
  - Octal O’777’
  - ASCII A’C’, ‘C’ (alternative)

PIC18 Instruction Format

- Instruction formats presented use 8-bits to specify a register file (f field)
  - Uses BSR to select only one bank at a time
  - When operating on a data register in a different bank, bank switching is needed
- Access Register
  - When operands are in the access bank, no bank switching is needed
  - Most SFRs are in the access bank

PIC18 Addressing Modes

- All MCUs use addressing modes to specify the operand to be operated on
- Register direct mode
  - Use an 8-bit value to specify a data register

```
MOVWF 0x25, A          MOVFF 0x40, 0x50
```

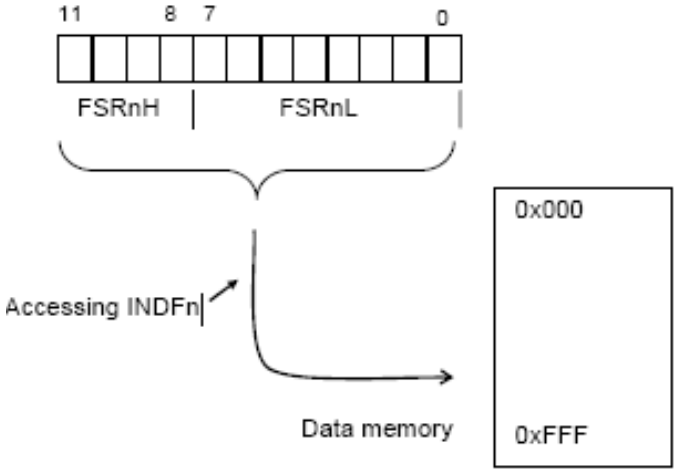
- Immediate mode
  - Actual operand provided, no need to access memory

```
MOVLW 0x25          ANDLW 0x40
```

PIC18 Addressing Modes

- Inherent mode
  - Operand is implied in the opcode field, opcode does not provide address
- Indirect mode
  - A special function register (FSRx) is used as a pointer to the actual data register

```
LFSR    FSR0, 0x25
MOVWF   INDF0
MOVWF   PREINC0
```



PIC18 Addressing Modes

- Bit-direct mode
  - Five instructions to deal with bits (BCF, BSF, BTFSC, BTFSS, BTG)

```
BTG PORTB, 2
```

PIC18 Instructions

- PIC18 has 77 instructions
- Data movement instructions

Mnemonic	Description	16-bit instruction word	Status affected
lfsr f, k	Load FSR	1110 1110 00ff k_{11}kkk 1111 0000 $k_7$kkk kkkk	None
movf f, d, a	Move f	0101 00da ffff ffff	Z, N
movff fs, fd	Move fs (source) to fd	1100 ffff ffff ffff 1111 ffff ffff ffff	None
movwf f, a	Move WREG to f	0110 111a ffff ffff	None
swapf f, d, a	Swap nibbles in f	0011 10da ffff ffff	None
movlb k	Move literal to BSR<3:0>	0000 0001 kkkk kkkk	None
movlw k	Move literal to WREG	0000 1110 kkkk kkkk	None

PIC18 Instructions

- Add instructions

Mnemonic	Description
addwf f, d, a	Add WREG and f
addwfc f, d, a	Add WREG and carry bit to f
addlw k	Add literal and WREG

- Subtraction instructions

Mnemonic	Description
subfwb f, d, a	Subtract f from WREG with borrow
subwf f, d, a	Subtract WREG from f
suwfb f, d, a	Subtract WREG from f with borrow
sublw k	Subtract WREG from literal

- References
  - Microchip DATASHEET
  - Huang, H., **PIC Microcontroller: An Introduction to Software & Hardware Interfacing**, Delmar Cengage Learning 2007