

<p>CLRW Clear W Register Szintaxis: [címke] CLRW Operandus: - Működés: 00h → W, 1 → Z Változó flag: Z Leírás: A W regiszter törlődik.</p>	<p>RRF Rotate Right f through Carry Szintaxis: [címke] RRF f,d Operandus: 0 ≤ f ≤ 127 d = 0 vagy 1 Működés: f<n> → dest<n-1> f<0> → C C → dest<7> Változó flag: C Leírás: A Carry flag-en keresztül jobbra lépteti az f regiszter tartalmát. Ha d=0, az eredmény a W, ha d=1, az f regiszterbe kerül.</p>
<p>CLRWDT Clear Watchdog Timer Szintaxis: [címke] CLRWDT Operandus: - Működés: 00h → (WDT), 0 → WDT előosztó, 1 → TO, 1 → PD Változó flag: TO, PD Leírás: Törli a Watchdog Timert, és – ha az hozzárendelt – az előosztóját.</p>	<p>SLEEP Szintaxis: [címke] SLEEP Operandus: - Működés: 00h → WDT, 0 → WDT előosztó, 1 → TO, 0 → PD Változó flag: TO, PD Leírás: Az oszcillátor leáll, a processzor SLEEP módba kerül.</p>
<p>COMF Complement f Szintaxis: [címke] COMF f,d Operandus: 0 ≤ f ≤ 127 d = 0 vagy 1 Működés: NEG(f) → (dest) Változó flag: Z Leírás: z f regiszter tartalma negálódik. Ha d=0, az eredmény a W, ha d=1, az f regiszterbe kerül.</p>	<p>SUBLW Subtract W from Literal Szintaxis: [címke] SUBLW k Operandus: 0 ≤ k ≤ 255 Működés: k - (W) → (W) Változó flag: C, DC, Z Leírás: tartalma kivonódik a k konstansból, az eredmény a W-be kerül.</p>
<p>DECF Decrement f Szintaxis: [címke] DECF f,d Operandus: 0 ≤ f ≤ 127 d = 0 vagy 1 Működés: (f) - 1 → (dest) Változó flag: Z Leírás: Az f regiszter tartalma dekrementálódik. Ha d=0, az eredmény a W, ha d=1, az f regiszterbe kerül.</p>	<p>SUBWF Subtract W from f Szintaxis: [címke] SUBWF f,d Operandus: 0 ≤ f ≤ 127 d = 0 vagy 1 Működés: (f) - (W) → (dest) Változó flag: C, DC, Z Leírás: Kivonja az f regiszter tartalmából a W tartalmát. Ha d=0, az eredmény a W, ha d=1, az f regiszterbe kerül.</p>
<p>DECFSZ Decrement f, Skip if Zero Szintaxis: [címke] DECFSZ f,d Operandus: 0 ≤ f ≤ 127 d = 0 vagy 1 Működés: (f) - 1 → (dest) Változó flag: - Leírás: Az f regiszter tartalma dekrementálódik. Ha az eredmény 0, a következő utasítást átlépi. Ha d=0, az eredmény a W, ha d=1, az f regiszterbe kerül. A ciklusszám: 1 vagy 2.</p>	<p>SWAPF Swap f Szintaxis: [címke] SWAPF f,d Operandus: 0 ≤ f ≤ 127 d = 0 vagy 1 Működés: f<0:3> → dest<4:7>, f<4:7> → dest<0:3> Változó flag: - Leírás: Az f regiszter alsó és felső négy bitjét felcseréli. Ha d=0, az eredmény a W, ha d=1, az f regiszterbe kerül.</p>
<p>GOTO Unconditional Branch Szintaxis: [címke] GOTO k Operandus: 0 ≤ k ≤ 2047 Működés: k → (PC) Változó flag: - Leírás: A k cím betöltődik a PC-be. A PC további bitjeinek kezelése típusonként változik. Ciklusszám 2.</p>	<p>XORLW Excl. OR Literal with W Szintaxis: [címke] XORLW k Operandus: 0 ≤ k ≤ 255 Működés: (W) XOR k → (W) Változó flag: Z Leírás: A W és a k konstans XOR kapcsolata. Az eredmény W-be kerül.</p>
<p>INCF Increment f Szintaxis: [címke] INCF f,d Operandus: 0 ≤ f ≤ 127 d = 0 vagy 1 Működés: (f) + 1 → (dest) Változó flag: Z Leírás: Az f regiszter tartalma inkrementálódik. Ha d=0, az eredmény a W, ha d=1, az f regiszterbe kerül.</p>	<p>XORWF Exclusive OR W with f Szintaxis: [címke] XORWF f,d Operandus: 0 ≤ f ≤ 127 d = 0 vagy 1 Működés: (W) XOR (f) → (dest) Változó flag: Z Leírás: A W és a f regiszter tartalmának XOR kapcsolata. Ha d=0, az eredmény a W, ha d=1, az f regiszterbe kerül.</p>
<p>INCFSZ Increment f, Skip if Zero Szintaxis: [címke] DECF f,d Operandus: 0 ≤ f ≤ 127 d = 0 vagy 1 Működés: (f) + 1 → (dest) Változó flag: - Leírás: Az f regiszter tartalma inkrementálódik. Ha az eredmény 0, a következő utasítást átlépi. Ha d=0, az eredmény a W, ha d=1, az f regiszterbe kerül. A ciklusszám: 1 vagy 2.</p>	<p>JELÖLÉSEK: f - fájlregiszter b - bitsorszám d - a művelet eredményének a helye (x) - x tartalma dest – eredmény helye</p>

Megjegyzések: