6 Arithmetic and Logic Instructions

- Unpacked = 4 0s out front
- Packed = no 0s out front, more effecient.
  - However, there are reasons for having 0000s out front
  - Could process nibble, instead of byte by byte
  - Makes processing easier
- BCD addition **does not** work.. kind of.
- `DA` takes into account the `AC` as well.
  - If `AC` $= 1$ or > 9, add $06H$
  - If `CY` $= 1$ or > 9, add $60H$.
- The extra `B` in `SUBB` mean subtracted with borrow.
  - Make sure to set clear `CY` before using `SUBB`
- One special case, divide by zero: `OV` $= 1$, values remain the same.
  - Division example: `A` $= 9$, `B` $= 5$
- `XRL` only works for 8 bits.
  - Same addressing modes as for `ANL`
- Compliment works for `A`, `C` or *anything* that is bit addressable.
- `CJNE` **changes the CY flag**
- Serial Communication example (we use RLC because we want to use the Carry flag for transmitting data)

```
ORG 0

MOV A, #35H
MOV P2, #0
MOV R0, #8
```

```
    SETB P2.1
    SETB P2.1

TX: RLC A
    MOV P2.1, C
    DJNZ R0, TX

    SETB P2.1
    SETB P2.1

    END
```

- Same example, backwards

```
    ORG 0

    MOV R0, #8

RX: MOV C, P2.5
    RRC A
    DJNZ R0, RX

    MOV R2, A

    END
```

Number of ‘1”s example

```
    ORG 0
    MOV R0, #0  ; Counter for 1s
    MOV R1, #8  ; Counter for loop

    MOV A, P2

LP: RRC A

    DJNE
```

$$n\text{bit} \cdot n\text{bit} = 2n\text{bit} \tag{1}$$

```
ORG 0
MOV R0, #30H
MOV @R0, #0

XCHD A, @R0      ; M[30] = 07, A = 30H
SWAP A           ; A = 30H
ORL A, #30H
```