Shift bits and throw carry away with ALR

In many cases when you shift/roll a byte to the right with LSR, you don't need the bits that are rolled out. So if you're planning on doing an ADC afterwards, you need a CLC inbetween.

```
lsr
clc
adc #$47
```

But with ALR you can AND the A register before it's shifted. So if you choose $fe (%11111110) as the AND mask, you set the bit that is rolled out to 0, and thereby making sure the carry is cleared, at no extra cost cyclewise.

```
alr #$fe
adc #$47
```

Unfortunately there is no equivalent for shifting the bits left. There is however a version for ROR called ARR. But this works more like ANC since it's setting the carry to the value of bit 7. So beware of the confusion this might cause.