Should you use Pull-ups?

TL;DR: AI offered some advice on configuration of unused pins, with outputs low or inputs as ADC to lower current drain. Personally, I didn't find it worth the gain and prefer the simplicity of pull-ups. It was fun to investigate and my results are here along with a quirk so please add any comments or wisdom.

A couple of spare pins on my 20M2 led me to think, whether there was any benefit to making them outputs, instead of assigning pull-ups. A quick google led to an interesting and plausible AI response, shown below:

	Power Draw	Safety (during debug)	EMI Protection
Output Low	Very Low	Moderate	Excellent
Input with Pull-up	Low	High	Good
Input as ADC	Lowest	High	Good
Floating	High	Low (High Risk)	Terrible

I was hooked! Outputs had lower power draw and lower EMI, although there's more outputs so more risk of accidental shorts. Or go one step further, set the pins as ADC inputs and get even lower power consumption by disabling the digital input circuitry without the output risk. Clearly I needed to play, and spoiler alert, it didn't save an awful lot power!

With 100 ohm ohm resistor on the +5.0V supply, I used a bench multimeter to measure the voltage drop and derive the supply current, later I replaced the resistor with 1k ohm for better range. The test program followed this template that consumed 9 bytes and different commands were inserted just above Main:

```
#picaxe 20M2
#no_data

'Nine REM blocks are placed here to show the 9 options tested. For test purposes each of these assumed all pins were not used.

Main:
    pause 3000
    goto Main
```

The program attached shows all the REM blocks for each of the options tried and details of the findings.

Based on a 20M2 , what I seem to have found is that:

1. Using the 'pullup $FFFF' command and default clock rate the current drain is about 701.8uA.
2. If pull-ups are not used the current drain is unstable and increased to 1.1mA, about +50%, and no doubt there's more EMI.
3. Setting all possible pins to ADC on this chip left 5 pins without pull-ups, so current was still unstable at at about 880uA.

4. Adding pullup to these pins reduced current to 701.1uA, about a 700nA improvement by using ADC inputs instead of pull-ups. Yes, thank you AI, but a very meagre saving compared to using pull-ups!!
5. Then a bizarre thing happened, if the pins were not set as ADC, as in ‘adcsetup = %0000000000000000’, or no adcsetup command. The current drain was the same whether I used ‘pullup $FFFF’, or ‘pullup %0111000110000000’ targeting just the non-ADC pins. The config is Option C3 in the attached, can anyone explain it?
6. For this PIC, contrary to AI, setting the pins as outputs achieved the lowest current drain, whether they were default (low), or set to low or high. However, it took about 20 seconds to settle down to 700.9uA. Marginally beating the ADC input approach, it saved about 900nA.
7. I had some 10k resistor arrays and using these as external pull-ups or pull-downs the current drain increased to about 705uA. Perhaps because of their lower resistance than the pull-up, which I calculated as about 37k ohm, when using a 120k pull-down resistor.
8. In terms of program space, a single pullup command controls all pins and is only 10 bytes, using the adcsetup adds 4 bytes, or choosing to set pins as output cost 2 bytes per pin.
9. For unused pins it seems pointless to use program bytes setting them as input because this is their default.

Unless you’re really scraping the bottom of the barrel, personally I think the KISS principle applies and pull-ups are the way to go. The 700nA saving with ADC inputs, or 900nA with outputs, was interesting. Power savings on other chips could be significant but for the PICAXE 20M2 it’s less than 0.13%, whereas you can save 33% by just shifting the clock rate down to 1MHz.

Hope you found it interesting. Feel free to share your preferences, corrections or wisdom. The quirky behaviour of settling the chip by just pulling up the non-ADC pins seems odd. Could it be just be some leakage current from other pull-ups?