BLINKENLIGHTS

(aka “Blinkie” or “Blinky”)

This document describes a ‘minimum parts count’ circuit to randomly flash nine (9) LEDs using only an RCA 1802 (CDP1802) CMOS 8-bit microprocessor and a minimum of support components (four parts besides the 1802 and the nine LEDs, not counting the battery or other power source).

The 1802 has several unique, or at least unusual for its time, features which lend themselves to this project:

- There are NO undefined instructions, so any data read on the data bus, regardless of bit pattern, can be interpreted by the 1802 as a valid instruction.
- The 1802 can operate statically, with a clock speed of as low as 0Hz, without malfunctioning (of course, if the clock is really 0Hz, then the 1802 just sits and waits for the next clock pulse before it does anything else). This means that when flashing LEDs at a rate slow enough to be perceptible by humans, the clock can be as slow as necessary, rather than having the clock be much faster (as most other microprocessors would require), making time delays in software necessary, which does not coincide with the plan of having a minimum parts count (hint – there would need to be a program, and memory to hold the program, etc).
- The 1802’s CMOS address lines are inherently current limited, with the range of currents being acceptable for operation of normal low power discrete LEDs. Thus, no extra resistors are required for current limiting, and no other parts are required for driving the LEDs.
- The 1802 is very low power, especially at low clock speeds, so even with some LEDs blinking on at any given moment, the entire circuit can operate for a long time on a set of batteries or other lower current power supply.
- The 1802’s clock circuit will work adequately with just one external inverter plus one resistor and one capacitor. In this Blinkenlights circuit, the external invertor is implemented using just one NPN bipolar transistor and one resistor.

Execution:

- The 1802 (the CPU) needs to have both of its positive supply pins (VDD on pin 40 and VCC on pin 16) connected to V+ of the power supply, and its common supply pin (aka “Ground” or VSS on pin 20) connected to the common/COM/GND of the power supply. To get a good level of illumination from the LEDs in this circuit’s configuration, the power supply voltage should be in the 4~6VDC range, and can be from a battery, a “wall wart” or similar power supply, a bench power supply, etc; The current is quite low, so even low-capacity batteries can power this circuit for an extended period of time.

- All 1802 control input pins (i.e. those that have a global affect on CPU operation) need to be connected to V+. These include WAIT (pin 2), CLR (pin 3), INT (pin 36), DMA-IN (pin 37), and DMA-OUT (pin 38). All of these inputs are active low, so having them always pulled up to V+ assures that none of these functions can be active, allowing simple and direct CPU operation to always be taking place.
- All of the "Data" bus I/O pins, D0~D7, are left floating so that random electrostatic fields surrounding the CPU can influence their logic states, EXCEPT it is critical to prevent the random occurrence of all Data bits being at logic LOW at the same time, which the CPU will interpret as a "HALT" command, which will stop program execution. Accordingly, it is important to connect ONE of the Data I/O pins to V+, and it is suggested that this be the D6 (pin 9) pin. Note that since the CPU is executing random instructions, it is possible that when executing those instructions which need to write data to the Data bus, that if the D6 pin is briefly configured as an output and also turned to a logic 0, that this will result in V+ being briefly tied to circuit ground via the D6 pin's internal circuit. The CPU can actually tolerate this brief and occasional occurrence. However, if you don't mind increasing the overall parts count by one (1), a pull-up resistor (perhaps 1k) can be placed between pin 6 and V+ rather than directly connecting them together, and you might feel better about it.
- The CPU clock needs to be correctly connected with its support components. The NPN transistor (a generic general-purpose type such as the common 2N2222 is appropriate) gets connected with its emitter (E) tied to circuit ground, and its collector (C) connected through a 100k pull-up resistor to V+; this makes an invertor. The invertor's resistor-collector junction (which was just described) must also be tied to the CPU's "CLK" input (pin 1). The CPU's "XTAL" output (pin 39) gets connected to one side of the capacitor (suggested value of 0.01µF), and the other side of this capacitor gets tied to the transistor's base (B), and also to one side of a 4.7M (not 4.7k) resistor, the other side of which gets connected to the CPU's "CLK" (pin 1), which already has things connected to it per the above instructions. With only the above connections made, the CPU can be powered up, and either pin 1 or pin 39 can be probed with an oscilloscope or logic probe to verify oscillation at roughly 20Hz (the frequency depends a lot on the tolerances of the components used in the oscillator circuit). Use a larger capacitance value (e.g. greater than 0.01µF) to make the clock run slower, resulting in each random pattern of LEDs being visible for a longer time before changing.
- The nine (9) LEDs can be connected to the CPU in various combinations. It is recommended to use the arrangement shown in this document's schematic. If you want to experiment with other patterns, keep these points in mind:
    - There are eight (8) address bus output pins available to work with (A0~A7, on pins 25~32 respectively), plus you can also use the CPU's "state code" output "SC0" output (pin 6) as per the schematic, and optionally also the CPU's "MWR" output (pin 35), along with the eight (8) address bus outputs. Possibly some other control outputs may be used as well, but for various reasons some of these will change state too briefly to be of use in this application.

- Each of the above address and SC0 pins can SOURCE current to a single LED and also SINK current from a single LED, so the anode of one LED *and* the cathode of another LED can be connected at any/all of the above-mentioned output pins. For the most part, all of the LEDs will be connected in this manner, in various arrangements, so that each LED is connected to one of the outputs at its anode, and to another of the outputs at its cathode. The schematic shows a typical arrangement that also includes the SC0 output pin, used in the same way as the address output pins. No pin should be connected to more than two LEDs.
- The LEDs can be any common lower power type, as have been used in electronic circuits for decades. It is recommended that all of the LEDs be the same color, make and model, in order to assure consistent illumination of each.

- If you experience problems with the circuit not starting up when power is applied, you might need to force the CPU to "reset" itself on power up. A reset is commanded to occur when the CPU's WAIT input (pin 2) is logically high (tied to V+) and its CLR input (pin 3) is normally high (V+) but briefly logically low (tied briefly to circuit ground). So, leave pin 2 connected to V+ and, instead of also always connecting pin 3 to V+, connect it as follows; connect another capacitor (recommended 0.1μF) between pin 3 and circuit ground, and also connect a pull-up resistor (recommended 10M) between pin 3 and V+. In this way, CLR pin 3 is normally pulled to logic high (V+) by the resistor, but when power is first applied (or re-applied after being off for at least a short while), the capacitor will momentarily pull CLR pin 3 logically low before allowing it to return to V+. This transient on the CLR input will cause the CPU to reset itself.

Description of Operation:

- When power is applied to the circuit, the CPU's clock will start running, and since all of the control inputs, which might otherwise put the CPU in some undesired state, are all pulled to V+ and thus disabled, the CPU will immediately start reading 'data' from the Data pins and trying to execute that 'data' as if it were actually valid instructions and associated data. The instructions executed might be commands to read from (or write to) some memory location, or to jump or branch to some other memory location, or perform some internal function such as dealing with internal registers or doing math or logic operations.
- Since any of the above instruction executions will necessarily increment the CPU's program counter, or in the case of jump or branch instructions, completely modify the program counter, the Address bus output pins will be changing their pattern on a regular basis, according to how fast the CPU clock is running.
- Since the 'data' being read from the unterminated/floating Data pins is random, the instructions and data the CPU is working with will make no sense, but the CPU does not know this or care, it just dutifully forges ahead with whenever spurious information it reads from the floating Data pins. Thus the pattern of bits on the Address output pins is also completely random.

- The LEDs are connected such that each of them depends on the two associated Address output pins (or the SC0 output pin) being at DIFFERENT logic levels. For example, if a given LED is connected between the A3 output pin and the A4 output pin, with its anode on A3 and its cathode on A4, then that LED will only illuminate when A3 is logically high (= V+) and A4 is logically low (= GND).
- OPTION – This document and the associated schematic show only nine (9) LEDs. A simple way to get even more LEDs going in additional random patterns is to connect several more LEDs, as many as nine more, to the already defined connection pattern per the schematic diagram. For example, per the previous paragraph, if you have an LED between A3 and A4 with the anode on A3 and the cathode on A4, a second LED can be connected in parallel with the first one, but with the opposite polarity so that its anode would be on A4 and its cathode on A3. Now, when A3 is high and A4 is low, the first LED will illuminate and the second LED will be off. But when A4 is high and A3 is low, the second LED will illuminate and the first LED will be off. When both of those Address pins are low, or both of them are high, both of the LEDs will be off. Because the A3 High and A4 Low state is completely random compared to the A3 Low and A4 High state, adding LEDs in this manner truly increases the number of random states and LED patterns.

Blinkenlights notice:

The following two of the many variations of a classic bit of computer whimsy, written in mangled mock German, that dates back to the days of the earliest kinds of computers……

ACHTUNG! ALLES LOOKENPEEPERS!

DAS COMPUTENMACHINE IS NICHT FÜR GEFINGERPOKEN UND MITTEN-GRABBEN. IST EASY SCHNAPPEN DER SPRINGENWERK, BLOWENFUSEN, UND POPPENCORKEN MIT SPITTZEN-SPARKEN. IST NICHT FÜR GEWERKEN BY DAS DUMMKOPFEN. DAS RUBBER-NECKEN SIGHTSEEREN KEEPEN HANDS IN DAS POKETS. RELAXEN UND WATCH DAS BLINKENLIGHTS.

****

ACHTUNG!

ALLES TURISTEN UND NONTEKNISCHEN LOOKENSPEEPERS!

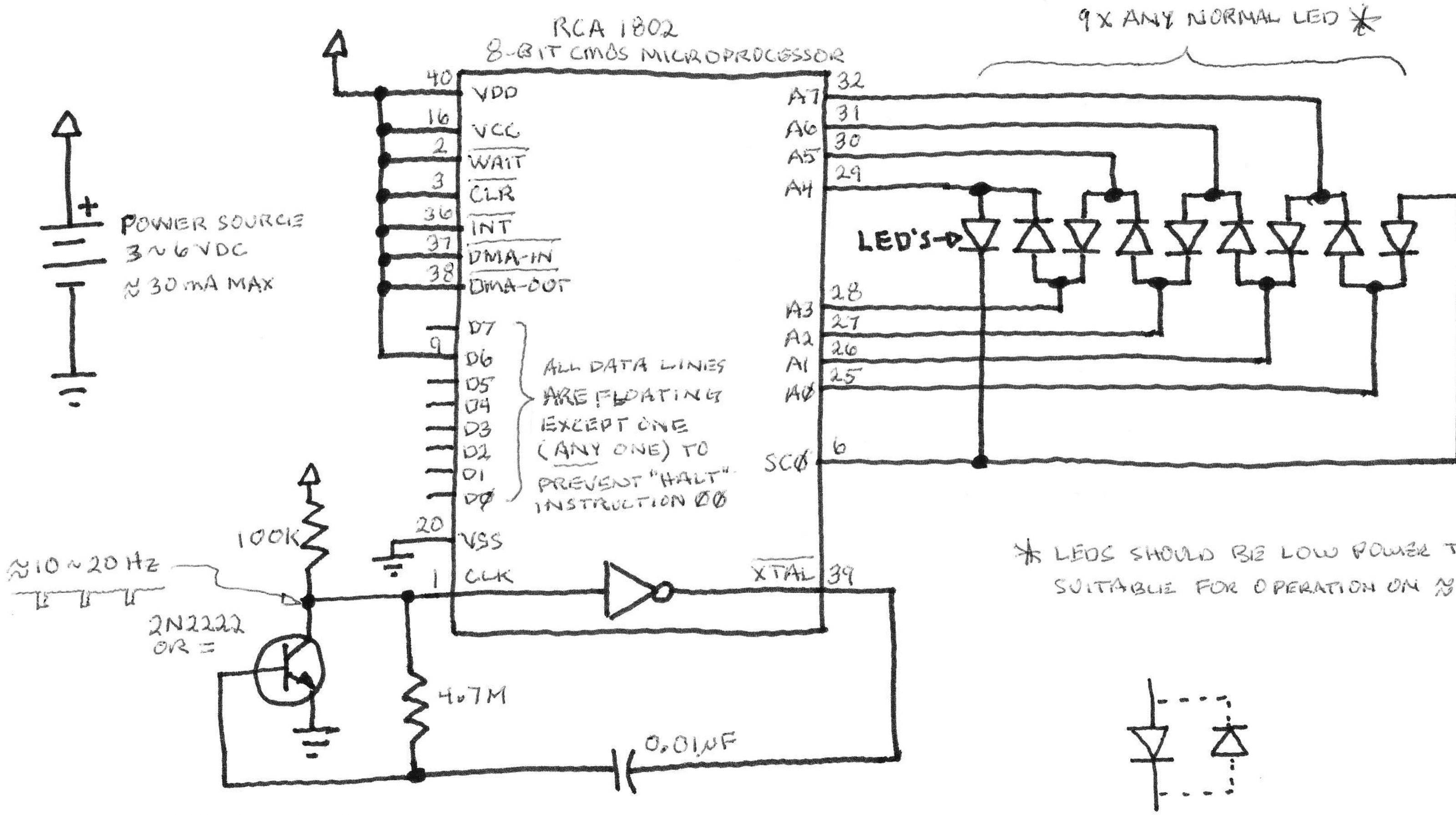
DAS KOMPUTERMASCHINE IST NICHT FÜR DER GEFINGERPOKEN UND MITTENGRABEN! ODERWISE IST EASY TO SCHNAPPEN DER SPRINGENWERK, BLOWENFUSEN UND POPPENCORKEN MIT SPITZENSPARKEN.

IST NICHT FÜR GEWERKEN BEI DUMMKOPFEN. DER RUBBERNECKEN SIGHTSEEREN KEEPEN DAS COTTONPICKEN HÄNDER IN DAS POCKETS MUSS.

ZO RELAXEN UND WATSCHEN DER BLINKENLICHTEN.

"BLINKENLIGHTS"

MINIMUM PARTS COUNT VERSION



* LEDS SHOULD BE LOW POWER TYPES SUITABLE FOR OPERATION ON ≈ 1mA

TRANSISTOR AND 100K RESISTOR ACT AS AN INVERTOR

OPTIONALLY, ADDITIONAL LED'S MAY BE CONNECTED IN REVERSE-PARALLEL TO THE LED'S SHOWN ABOVE, GREATLY INCREASING THE NUMBER OF RANDOM PATTERNS

BASED ON ORIGINAL DESIGN BY TECHNICAL MICRO SYSTEMS INC

REDRAWN 2023 BY PAUL SCHMIDT