

Circuitos de redstone

Um **circuito de redstone** é uma construção focada principalmente na manipulação de sinais de redstone e o uso de lógica. Eles podem ser usados para transmitir sinais de redstone, realizar operações lógicas (AND, OR etc.), armazenar dados, diminuir ou aumentar sinais de redstone, construir relógios e mais.

Uma distinção útil pode ser feita entre um **circuito** realizando operações em sinais (gerando, modificando, combinando etc.) e um **mecanismo** manipulando o ambiente (movendo blocos, abrindo portas, alterando o nível de luz, produzindo sons etc.). Criar esta distinção nos permite falar sobre os vários circuitos de maneira separada, além da livre escolha dos jogadores para o que desejarem fazer. As máquinas controladas por sinais de redstone podem variar desde dispositivos simples, como portas automáticas e interruptores, até dispositivos complexos, como elevadores, *farms* automáticas, ou até computadores dentro do jogo. Entretanto, *este* artigo aborda apenas uma visão geral dos *circuitos* de redstone antes citados. Estes podem ser utilizados para controlar mecanismos simples ou de maneira combinada como parte de uma construção maior. Cada tipo de circuito nesta página possuem links para a sua própria, o que os cobrem em maiores detalhes e fornece esquemas para as suas múltiplas variações.

Antes de lidar com qualquer um dos mais básicos circuitos de redstone, é necessário a compreensão de conceitos básicos, como: "energia", "força do sinal", "ticks de redstone" e "atualizações de bloco". Alguns artigos relevantes estão listados abaixo:

- O artigo sobre as [mecânicas de redstone](#) aborda mais informações sobre estes conceitos.
- O artigo sobre os [componentes de redstone](#) aborda todos os blocos que interagem com a energia de redstone como um todo através de listas com descrições.
- O tutorial sobre os [tipos de mecanismos](#) complementa este artigo com uma variedade de designs usando os circuitos descritos aqui.
- O tutorial sobre [dicas com a redstone](#) dá conselhos gerais sobre a construção de engenhocas de redstone.



Descrevendo os circuitos

A maioria dos circuitos são descritos usando os diagramas de [esquemas](#); alguns destes requer que múltiplas imagens apareçam em uma ou duas camadas por imagem. Veja a página de [ajuda](#) para detalhes em como os vários blocos e componentes são representados.

Tamanho

A wiki descreve o tamanho de um circuito (o volume do sólido retangular que ele ocupa) com a notação: *profundidade × largura × altura*, incluindo os blocos de suporte/base, mas sem incluir as entradas e saídas.

Outro método usado para descrever o tamanho de um circuito na comunidade de *Minecraft* é ignorar os blocos que não são de redstone simplesmente usados para suporte (como, por exemplo, os blocos abaixo do pó de redstone ou de repetidores). Entretanto, este método é incapaz de distinguir entre circuitos retos e de 1 de altura, além de algumas outras diferenças de circuito.

Às vezes é mais conveniente comparar circuitos simplesmente pela área de suas pegadas (por exemplo: 3×4 para um circuito de três blocos de altura por quatro blocos de largura), ou por uma única dimensão importante num contexto específico (por exemplo: a largura numa sequência de sub-circuitos, a altura num espaço pequeno etc.).

Recursos

Diversos recursos podem ser considerados como objetivos de design desejáveis:

1 de altura Um circuito que possui um de altura é aquele em que sua dimensão vertical é de apenas um bloco de altura (significando que não pode haver nenhum componente de redstone que requer blocos de suporte abaixo deles, como o pó de redstone, os repetidores etc.). Veja também os retos.

1 de largura Um circuito que possui um de largura é aquele em que sua dimensão horizontal é de exatamente um bloco de largura.

Reto Um circuito reto é aquele que geralmente pode ser colocado no chão sem componentes acima de outros (não há problema com blocos de suporte abaixo de componentes). As estruturas retas são normalmente mais fáceis para os iniciantes entenderem e construírem, e seu tamanho é ótimo para ficar abaixo do chão ou no topo de telhados. Veja também 1 de altura.

Nivelado Um circuito nivelado é aquele que não se estende para além de uma parede reta, chão ou teto e ainda pode ser útil do outro lado, embora mecanismos de redstone possam ser visíveis na parede. Este é um objetivo de design para os que gostam de usar extensores e portas de pistões etc. Veja também hipster e escondidinho.

Hipster Um circuito hipster é aquele que começa escondido atrás de uma parede reta, chão ou teto e ainda pode ser útil do outro lado. Veja também nivelado e escondidinho.

Instantâneo Um circuito instantâneo é aquele em que sua saída responde imediatamente à entrada (sem atrasos).

Escondidinho Um circuito "escondidinho" é aquele onde não há componentes de redstone visíveis antes e depois de sua tarefa ser completa (mas sem problemas caso apareçam durante a operação). Este é um objetivo de design desejável para os que gostam de usar extensores e portas de pistões etc. Veja também nivelado e hipster.

Silencioso Um circuito silencioso é aquele que não faz barulho (como os sons dos pistões se movendo, dos ejetores/liberadores ativando quando vazios etc.). As estruturas silenciosas são desejáveis para armadilhas ou lares pacíficos.

Empilhável Um circuito empilhável é aquele que pode ser colocado "diretamente" no topo de suas outras cópias, com todos eles podendo ser controlados por uma única unidade. Veja também padronizado.

Expandível Um circuito expandível é aquele que pode ser colocado "diretamente" próximo de suas outras cópias, com todos eles podendo ser controlados por uma única unidade. Veja também padronizado.

Padronizado Um circuito padronizado é aquele que pode ser colocado "diretamente" próximo ou no topo de suas outras cópias, com cada cópia ainda podendo ser controlada de maneira independente. Veja também empilhável.

Circuitos podem ser descritos como "padrões de dois de largura" (padronizado a cada dois espaços em uma dimensão), ou "2×4 padronizado" (em duas direções) etc. Algumas estruturas podem ser descritas como um "padrão alternável", significando que eles podem ser colocados próximos de si caso qualquer outro esteja virado ou com um design ligeiramente diferente.

Outros objetivos de design podem incluir reduzir o atraso que um sub-circuito adiciona a um circuito maior, reduzir o uso de componentes caros em recursos (redstone, quartzo do Nether etc.) e reorganizar ou refazer um circuito para torná-lo o menor possível.

Alguns componentes não estão disponíveis até que um jogador obtenha acesso ao Nether, o que limita os designs disponíveis. Em particular, os comparadores de redstone, os observadores e os detectores de luz solar requerem quartzo do Nether, do qual está disponível apenas no Nether. Além disso, as lâmpadas de redstone requerem pedra luminosa, que se torna ocasionalmente disponível através do comércio ou das bruxas, mas encontrado em maior abundância no Nether.

Tipos de circuitos

Embora

o número de maneiras para construir circuitos seja infinita, certos padrões de construção ocorrem repetidamente. As seguintes seções tentam categorizar aqueles que se provaram ser úteis para a comunidade de *Minecraft*, enquanto que os artigos principais descrevem os circuitos específicos que caem nestas categorias.

Alguns destes circuitos podem ser usados por eles mesmos para o controle simples de mecanismos, mas é necessário que o jogador frequentemente os combinem em circuitos mais complexos para atender as necessidades de um mecanismo.

Circuito de transmissão

Artigo principal: Circuito de transmissão

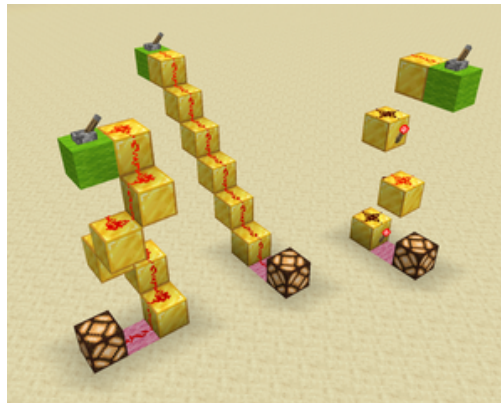
Alguns aspectos da transmissão de sinal podem ser úteis para entender: os tipos de transmissão, a transmissão vertical, os repetidores e os diodos.

Transmissão vertical

Transmissão de sinais de baixo para cima.



Transmissão de sinais de cima para baixo.



Exemplos de escadas verticais na *Edição Bedrock*.

Note como o vidro facilita a transmissão em ambas as direções, mas os funis permitem com que os sinais sejam enviados apenas de baixo para cima.



Embora a transmissão horizontal de sinal seja direta, a transmissão vertical envolve opções e compensações.

- **Escadas de redstone:** A maneira mais simples, mas não a melhor, de transmitir sinais verticalmente é colocar pó de redstone em blocos de baixo para cima na diagonal, tanto numa escada reta de

blocos, num espiral 2×2 de blocos, ou em outra variação similar. As escadas de redstone podem transmitir sinais tanto de baixo para cima como de cima para baixo, mas ocupam espaço demais e requerem repetidores a cada 15 blocos. Como a pedra luminosa, o topo das lajes, o vidro e escadas viradas de cabeça para baixo podem suportar pó de redstone mas não cortando-o, os sinais podem ser transmitidos verticalmente (de baixo para cima apenas) ao alternar estes blocos em uma "escada de mão" 2×1. As escadas de mão de redstone ocupam menos espaço do que as escadas de redstone, mas também requerem repetidores a cada 15 blocos. Na Edição Bedrock, os vidros e pistões podem ser usados para criar escadas de mão verticais de dois caminhos que transmitem sinais tanto de baixo para cima como de cima para baixo (pedra luminosa, funis e lajes ainda permitem com que o pó energize de baixo para cima mas não de cima para baixo).

- **Torres de tochas e escadas de tochas:** Uma tocha de redstone pode energizar o bloco acima dela ou o pó de redstone abaixo dela, permitindo a transmissão vertical tanto de baixo para cima como de cima para baixo (é necessário designs diferentes para cada). Como cada tocha requer um certo tempo para alternar de estado, uma torre de tochas pode introduzir um certo atraso no circuito, sem que repetidores sejam necessários. Entretanto, cada tocha inverte o sinal de redstone (por exemplo: saindo de energizado para não energizado), então possuir um número par de tochas é necessário.

- **Torres de observadores:** Um observador pode energizar um bloco de circuito de redstone acima ou abaixo dele, permitindo a transmissão vertical tanto de baixo para cima como de cima para baixo. Colocar blocos que podem ser ativados, como o pó de redstone, blocos musicais ou portas, ambos, em cima e em baixo, criam uma mudança de estado para o bloco acima quando o observador está olhando para baixo ou para o bloco abaixo quando o observador está olhando para cima. Repetir esse padrão significa que as atualizações serão feitas em cadeia.

- **Explorando o detector de luz solar:** Você pode usar detectores de luz solar para enviar um sinal de redstone para baixo em 1 tick, mas o caminho não pode ser obstruído por nada. Você precisa ter um pistão empurrando um bloco na frente do sensor. Ele detecta a mudança na luz e emite um pulso de redstone. Este design é pode ser estendido para cima o quão longe desejar, mas você precisa ter o buraco original aberto à luz do sol. Ele também funciona apenas durante o dia, já que usa as sombras para ativar.

- **Colunas de bolhas:** Um observador pode ser usado para detectar a atualização de bloco que ocorre quando a fonte de água se torna uma coluna de bolhas (ou vice-versa). Ao substituir o bloco abaixo de uma coluna de fontes de água para areia das almas ou um bloco de magma, a coluna inteira imediatamente se torna blocos de coluna de bolhas. Isso pode ser usado para transmitir rapidamente um sinal de redstone de

baixo para cima para um observador olhando para o topo do bloco de fonte de água ou coluna de bolhas.

- **Atualizando o muro:** Uma configuração que pode carregar um pulso de sinal de cima para baixo entre qualquer distância envolvem muros

de qualquer tipo, um pistão e um observador. Quando um bloco de muro possui um bloco sólido nos dois lados opostos e blocos não sólidos (por exemplo: o ar) nos outros dois lados, um formato reto se forma. Isso pode ser repetido verticalmente para cima até qualquer altura.

Entretanto, quando um bloco de muro ou sólido é colocado dentro de outros dois blocos de ar em torno de uma parede reta, o bloco de parede *reta e todo bloco de parede reta abaixo dele* serão atualizados para uma versão diferente da parede com uma coluna no meio. Essa atualização é instantânea e pode ser detectada por um observador observando qualquer parede reta na torre. A atualização pode ser repetida ao possuir um pistão olhando para a parede reta no topo da torre, já que a cabeça do pistão também aciona a atualização da parede.

- **Escadas de redstone:** A maneira mais simples, mas não a melhor, de transmitir sinais verticalmente é colocar pó de redstone em blocos de baixo para cima na diagonal, tanto numa escada reta de blocos, num espiral 2x2 de blocos, ou em outra variação similar. As escadas de redstone podem transmitir sinais tanto de baixo para cima como de cima para baixo, mas ocupam espaço demais e requerem repetidores a cada 15 blocos.
- **Torres de tochas e escadas de tochas:** Uma tocha de redstone pode energizar o bloco acima dela ou o pó de redstone abaixo dela, permitindo a transmissão vertical tanto de baixo para cima como de cima para baixo (é necessário designs diferentes para cada). Como cada tocha requer um certo tempo para alternar de estado, uma torre de tochas pode introduzir um certo atraso no circuito, sem que repetidores sejam necessários. Entretanto, cada tocha inverte o sinal de redstone (por exemplo: saindo de energizado para não energizado), então possuir um número par de tochas é necessário.
- **Torres de observadores:** Um observador pode energizar um bloco de circuito de redstone acima ou abaixo dele, permitindo a transmissão vertical tanto de baixo para cima como de cima para baixo. Colocar blocos que podem ser ativados, como o pó de redstone, blocos musicais ou portas, ambos, em cima e em baixo, criam uma mudança de estado para o bloco acima quando o observador está olhando para baixo ou para o bloco abaixo quando o observador está olhando para cima. Repetir esse padrão significa que as atualizações serão feitas em cadeia.
- **Explorando o detector de luz solar:** Você pode usar detectores de luz solar para enviar um sinal de redstone para baixo em 1 tick, mas o caminho não pode ser obstruído por nada. Você precisa ter

um pistão empurrando um bloco na frente do sensor. Ele detecta a mudança na luz e emite um pulso de redstone. Este design é pode ser estendido para cima o quão longe desejar, mas você precisa ter o buraco original aberto à luz do sol. Ele também funciona apenas durante o dia, já que usa as sombras para ativar.

- **Colunas de bolhas:** Um observador pode ser usado para detectar a atualização de bloco que ocorre quando a fonte de água se torna uma coluna de bolhas (ou vice-versa). Ao substituir o bloco abaixo de uma coluna de fontes de água para areia das almas ou um bloco de magma, a coluna inteira imediatamente se torna blocos de coluna de bolhas. Isso pode ser usado para transmitir rapidamente um sinal de redstone de baixo para cima para um observador olhando para o topo do bloco de fonte de água ou coluna de bolhas.
- **Atualizando o muro:** Uma configuração que pode carregar um pulso de sinal de cima para baixo entre qualquer distância envolvem muros de qualquer tipo, um pistão e um observador. Quando um bloco de muro possui um bloco sólido nos dois lados opostos e blocos não sólidos (por exemplo: o ar) nos outros dois lados, um formato reto se forma. Isso pode ser repetido verticalmente para cima até qualquer altura. Entretanto, quando um bloco de muro ou sólido é colocado dentro de outros dois blocos de ar em torno de uma parede reta, o bloco de parede *reta e todo bloco de parede reta abaixo dele* serão atualizados para uma versão diferente da parede com uma coluna no meio. Essa atualização é instantânea e pode ser detectada por um observador observando qualquer parede reta na torre. A atualização pode ser repetida ao possuir um pistão olhando para a parede reta no topo da torre, já que a cabeça do pistão também aciona a atualização da parede.

Repetidor"Repetir" um sinal significa impulsioná-lo de volta à força máxima. A maneira mais fácil de fazer isso é com um repetidor de redstone. As variações incluem:

- Repetidor instantâneo: Repete um sinal sólido sem o atraso introduzido pelo repetidor de redstone.
- Repetidor de dois caminhos: Repete um sinal em ambas as direções.
- Repetidor instantâneo: Repete um sinal sólido sem o atraso introduzido pelo repetidor de redstone.
- Repetidor de dois caminhos: Repete um sinal em ambas as direções.

ComparadorUm comparador possui dois modos: comparação e subtração. O modo de comparação compara a força do sinal do lado com a entrada; caso o sinal do lado seja mais forte do que o sinal de entrada, a saída é desligada e, caso contrário, é ligada. No modo de subtração, o comparador subtrai o sinal do lado da entrada.

DiodoUm "diodo" é um circuito de caminho único que permite com que um sinal trafegue uma única direção. É usado para proteger outro circuito de sinais tentando invadir a saída, o que poderia mudar incorretamente o

estado do circuito ou interferir o seu temporizador. Também é usado em um circuito compacto para evitar com que uma parte do circuito interfira o outro. As escolhas comuns para um diodo incluem um repetidor de redstone ou uma elevação de altura até uma pedra luminosa ou o topo de uma laje, que não transmite o sinal de volta para baixo.

Vários circuitos já são de caminho único simplesmente devido ao fato de que suas saídas vêm de um bloco que não pode receber entradas. Por exemplo, um sinal não pode ser empurrado de volta para um circuito através de uma tocha de redstone exceto através do bloco em que está anexado.

Circuito lógico

Artigo principal: Circuito lógico

Um **portão lógico** é um circuito que recebe um ou mais entradas e produz uma única saída quando alguma condição é atendida. A força do sinal de entrada nestes circuitos normalmente não importa. Quando uma entrada ou saída fornece o sinal de redstone, ele é normalmente só referido como "ligado" e, quando não há sinal, é referido como "desligado". Por exemplo, um portão AND leva dois sinais de redstone como entrada e devolve um sinal de redstone apenas se ambas as entradas forem maiores que 0. Isto é, um portão AND só é ligado quando ambas as entradas estão ligadas.

Em diagramas eletrônicos ou de programação, portões lógicos são tipicamente exibidos como se fossem dispositivos individuais; porém, ao construir dispositivos de redstone no *Minecraft*, todos os portões lógicos são formados de múltiplos blocos e componentes, que interagem entre si para produzir os resultados esperados.

Portão NOT

Um portão NOT (também conhecido como "inversor") é ligado caso a sua entrada esteja desligada. O portão NOT mais simples é um sinal de entrada com uma tocha de redstone anexada.

Portão OR

Um portão OR só é ligado caso *qualquer uma* de suas entradas estiverem ligadas. O portão OR mais simples é alimentar múltiplos sinais de entrada juntos num único bloco ou fio de redstone.

Portão NOR

Um portão NOR só é ligado caso *nenhuma* de suas entradas estejam ligadas. O portão NOR mais simples é alimentar múltiplos sinais de entrada num bloco com uma tocha de redstone anexada.

Portão AND

Um portão AND só é ligado caso *todas* as suas entradas estiverem ligadas. O portão AND mais simples é adicionar tochas de

redstone na frente de cada entrada em um portão NOR.

Portão NAND

Um portão NAND só é ligado caso *qualquer uma* de suas entradas estejam desligadas. O portão NAND mais simples é remover a última tocha de redstone de um portão AND.

Portão XOR

Um portão XOR é ligado quando as suas entradas são *diferentes*.

Portão XNOR

Um portão XOR é ligado quando as suas entradas são *iguais*.

Portão IMPLY

Um portão IMPLY é ligado a menos que a primeira entrada esteja ligada e a segunda entrada esteja desligada.

A	Ligado	Ligado	Desligado	Desligado	Pergunta respondida
B	Ligado	Desligado	Ligado	Desligado	
<u>A AND B</u>	Desligado	Desligado	Desligado	Desligado	A e B estão ligados?
<u>NOT (A IMPLIES B)</u>	Desligado	Desligado	Desligado	Desligado	A e B estão desligados?
<u>NOT (B IMPLIES A)</u>	Desligado	Desligado	Desligado	Desligado	B e A estão desligados?
<u>A NOR B</u>	Desligado	Desligado	Desligado	Desligado	Ambas as entradas estão desligadas?
A	Desligado	Desligado	Desligado	Desligado	A está ligado?
<u>A XOR B</u>	Desligado	Desligado	Desligado	Desligado	As entradas são diferentes?
<u>NOT A</u>	Desligado	Desligado	Desligado	Desligado	A está desligado?
<u>A XNOR B</u>	Desligado	Desligado	Desligado	Desligado	As entradas são as mesmas?
B	Desligado	Desligado	Desligado	Desligado	B está ligado?
<u>NOT B</u>	Desligado	Desligado	Desligado	Desligado	B está desligado?
<u>A NAND B</u>	Desligado	Desligado	Desligado	Desligado	Ambas as entradas estão desligadas?
<u>A IMPLIES B</u>	Desligado	Desligado	Desligado	Desligado	Se A está ligado, B também está?
<u>B IMPLIES A</u>	Desligado	Desligado	Desligado	Desligado	Se B está ligado, A

A	Ligado	Ligado	Desligado	Desligado	Pergunta respondida
					também está?
<u>A OR B</u>	Desligado	Desligado	Desligado	Desligado	Ambas as entradas estão ligadas?

Circuito de pulso

Artigo principal: [Circuito de pulso](#)

Um **circuito de pulso** produz um sinal de redstone por um período específico de tempo, alterando a duração do sinal de entrada ou reagindo aos sinais de duração particular. Outros circuitos ou mecanismos podem requerer pulsos de duração específica para operarem corretamente, ou pode usar a duração do pulso para transmitir informações.

Um circuito que é estável em um estado de saída e instável no outro é conhecido como um [circuito monoestável](#).^{[[nota 1](#)]}

Vários circuitos de pulso são monoestáveis porque o seu estado desligado é estável, mas o seu estado ligado logo volta a ser desligado.

[Gerador de pulso](#) Um gerador de pulso produz um pulso de duração específica.

[Limitador de pulso](#) Um limitador de pulso (também conhecido como "encurtador de pulso") reduz a duração de pulsos que são longos demais.

[Extensor de pulso](#) Um extensor de pulso (também conhecido como "sustentador de pulso" ou "aumentador de pulso") aumenta a duração de pulsos que são curtos demais.

[Multiplicador de pulso](#) Um multiplicador de pulso devolve múltiplos pulsos para cada pulso de entrada (multiplica o número de pulsos).

[Divisor de pulso](#) Um divisor de pulso (também conhecido como "contador de pulso") devolve o sinal apenas após uma certa quantidade de pulsos serem detectados através da entrada (o número de pulsos é indicativo ao número de repetições).

[Detector de borda](#) Um detector de borda reage tanto ao sinal de redstone alterando de desligado para ligado (um detector "ascendente"), de ligado para desligado (um detector "descendente"), ou alternando entre ligado e desligado em qualquer ordem (um detector de "duas fases").

[Detector de largura de pulso](#) Um detector de largura de pulso reage apenas aos pulsos num certo raio de duração (frequentemente apenas aos pulsos de duração única e específica).

Circuito de relógio

Artigo principal: [Circuito de relógio](#)

Um circuito de relógio é um gerador de pulso que produz uma repetição de pulsos específicos repetidamente. Alguns são projetados para funcionarem para sempre, enquanto que outros podem ser parados e iniciados.

Um relógio simples com apenas dois estados de durações iguais é nomeado pela duração do seu estado ligado (por exemplo: um relógio que alterna entre um estado ligado e desligado de 5 ticks é chamado de "relógio 5"), enquanto que outros são normalmente nomeados pelo seu período (o tempo que leva para que o relógio voltar ao seu estado original; por exemplo, um "relógio de 1 minuto" pode produzir um pulso de 1-tick a cada 60 segundos).

1º relógio observadorUm relógio de repetição feito com observadores e pistões (um observador olhando para um pistão).

2º relógio observadorUm relógio de repetição feito com dois observadores olhando para si mesmos.

Relógio de repetidorUm relógio de repetidor consiste em uma repetição de repetidores (normalmente repetidores de redstone ou tochas de redstone) com pós ou blocos ocasionais para extrair os pulsos apropriados.

Relógio de funilUm relógio de funil produz pulsos temporizados ao mover itens para lá e para cá entre dois funis alimentando-se e levando a saída de redstone com comparadores.

Relógio de pistãoUm relógio de pistão produz uma repetição de pulsos ao passar um bloco para trás e para frente (ou ao redor, com vários pistões) e extraíndo o pulso de redstone quando o bloco está numa certa localização.

Relógio de comparadorÉ um relógio de duração de ciclo curto ou moderado utilizando a subtração do comparador ou o recurso de esvanecimento de sinal. Os relógios também podem ser construídos usando detectores de luz solar, carrinhos de mina, botes, fluxo de água, desaparecimento de itens etc.

Circuito de memória

Artigo principal: Circuito de memória

Os circuitos de memória mantêm as suas saídas até que recebam um sinal que os digam para alterar sua saída. Em termos simples, eles permanecem ligados ou desligados até que recebam um sinal dizendo-lhes para desligar ou ligar. Isso significa que um estado de circuito de memória (a saída) depende da entrada atual e das entradas anteriores. A maioria dos outros circuitos apenas geram temporariamente uma saída baseada na entrada atual.

A maioria dos circuitos de memória são chamados de "*latches*" (travas) ou "*flip-flops*"; estes são nomeados através de circuitos eletrônicos reais pois eles se comportam de maneira similar. Alguns tipos básicos de circuitos de memória estão descritos abaixo. Veja aqui para uma explicação dos termos técnicos.

Os circuitos de memória são nomeados baseados nos tipos de entrada que recebem e quaisquer portões lógicos que utilizam.

Trava DR Uma trava DR possui duas entradas: a que **define** (D) o circuito, ligando-o, e a que **reinicia** (R) o circuito, desligando-o. O circuito de memória mais velho e mais comum no *Minecraft* é a trava DR construída com um portão NOR (trava DR NOR).

A flip-flop Um A *flip-flop* possui uma entrada: a que **alterna** (A), mudando a saída do circuito de ligado para desligado, ou de desligado para ligado.

Trava com portão D Uma trava com portão D possui duas entradas: o **relógio** (R), que determina quando o circuito pode alterar a sua saída, e o **dado** (D), que determina a saída. É chamado de "portão" pois o relógio age como um, permitindo que o circuito atualize dependendo do estado do relógio.

Trava JK Uma trava JK possui duas entradas: J, que liga o circuito e K que desliga o circuito; se ambos estiverem ligados, então o circuito alterna a sua saída. Estes circuitos também frequentemente possuem uma entrada de relógio (R) que determina quando o circuito pode alterar a sua saída. (J e K não possuem nomes comuns como "define", "reinicia", "alterna" etc. antes vistos).

Contador Ao contrário do A *flip-flop* e a trava DR, que pode armazenar dois estados (ligado e desligado), um contador é projetado para armazenar um número grande de estados.

Vários outros circuitos de memória são possíveis.

Circuitos de pistão

Artigo principal: [Circuitos de pistão](#)

Os pistões

permitem com que os jogadores criem circuitos que são pequenos e/ou mais rápidos do que suas contrapartes padrões e feitas apenas com redstone. Uma compreensão dos circuitos de redstone padrões pode ajudar, já que este tutorial é focado no design do circuito em vez da função. Os principais componentes aqui são os pistões grudentos, pistões normais, fio de redstone, repetidores e tochas de redstone.

Há vários benefícios de se fazer um circuito com pistões:

- Nem os repetidores nem os pistões "apagam", como é o caso das tochas de redstone.
- Os circuitos de pistão são frequentemente (não sempre) pequenos e/ou mais rápidos do que suas contrapartes feitas de redstone. Isso permite construir dispositivos como relógios rápidos e transmissão "instantânea" de sinal.
- A habilidade do pistão de mover blocos no mundo os tornam naturais para circuitos de memória, além das portas e pontes alternáveis. Com blocos de slime ou mel envolvidos, estruturas inteiras podem "subir e se mover" (veja também o [tutorial sobre máquinas voadoras](#)).

- Os circuitos de pistão podem reduzir consideravelmente o uso de redstone a favor de madeira, pedra e ferro.

Circuitos diversos

Artigo principal: [Circuitos diversos](#)

Estes circuitos geralmente não são necessários para projetos de redstone, mas podem ser úteis para projetos complexos, provas de conceito e experimentos. Alguns exemplos:

Multiplexos e relays Um multiplexo é uma forma avançada de portão lógico que escolhe entre duas entradas para deixar passar como uma entrada baseado em uma entrada adicional (por exemplo: se a entrada A está ligada, então devolve a entrada B, caso contrário, devolve a entrada C). O reverso disso é o relay, que copia o dado da entrada para uma das duas saídas, dependendo se a entrada adicional está ligada ou desligada.

Aleatorizadores

Artigo principal: [Tutorial:Como fazer aleatorizadores no Minecraft](#)

Um aleatorizador produz sinais de saída imprevisivelmente. Eles podem ser projetados para produzir um pulso em intervalos aleatórios, ou para aleatorizar quais das múltiplas saídas estão ligadas (como geradores de números aleatórios ou RNGs). Alguns aleatorizadores usam a natureza aleatória do *Minecraft* (como o crescimento de [cactos](#) ou a seleção de espaço do [ejetor](#)), enquanto outros produzem pseudo-aleatoriedades algoritmamente.

Circuitos multi-bit Circuitos multi-bit tratam as suas linhas de entrada como um valor multi-bit único (algo além de zero e um) e realiza uma operação em todos eles de uma vez. Com tais circuitos, possivelmente combinado com matrizes de circuitos de memória, é possível construir calculadoras, relógios digitais, e até computadores básicos dentro do *Minecraft*.

Detectores de atualização de blocos

Artigo principal: [Tutorial:Como fazer um detector de atualização de bloco no Minecraft](#)

Artigo principal: [Tutorial:Como fazer um detector de atualização de comparador no Minecraft](#)

Um detector de atualização de bloco (DAB) é um circuito que reage a um bloco alterando de estado (por exemplo: uma pedra sendo minerada, a água virando gelo, uma abóbora próxima a um caule etc.). Os DABs reagem produzindo um pulso, enquanto que os A-DABs (DABs alternáveis) reagem ao alternar o seu estado de saída. Estes são geralmente baseados em quirks sutis ou travamentos no comportamento do dispositivo; os circuitos atuais frequentemente dependem de pistões. A partir da [Edição Java 1.11](#), a maioria das funções dos DABs foram condensadas dentro do [observador](#),

entretanto, um circuito DAB também pode detectar outras mudanças que não são detectáveis pelos observadores, como uma fornalha terminando de

fundir ou algo sendo fabricado na bancada. A adição disso foi feita para buscar uma paridade com as versões da Edição Bedrock.

Circuitos mais avançados

Artigo principal: Tutorial:Como fazer circuitos avançados de redstone no Minecraft

Vários outros circuitos complexos são possíveis.