

# Comment comptent les eaurdinateurs... ...et les humains

*Ceci est le texte animateur "parlé" correspondant à la présentation complète → ~45mn et aux fiches légèrement allégées → peut se réduire à ~15mn. Cf [dossier](#).*

*Les sources des images sont indiquées dans le champ popup.*

*Alternier la lecture avec participations de la salle (à minima, poser des questions à la salle).*

*L'ordinateur à eau peut être imprimé en 3D ([source](#)). A remplir à la seringue (ex: mouche-bébé) avec de l'eau + colorant alimentaire. Mettre l'ordinateur à plat, seringuer les bits de biais vers le centre des cases à remplir aux  $\frac{3}{4}$ , puis mettre verticalement pour le calcul. Bien secouer pour vider les bulles avant réusage.*

## Intro:

On va vous montrer comment marche notre "eaurdinateur" à eau (montrer), et plus généralement, comment font les ordinateurs pour calculer. Comme les ordinateurs sont un peu bêtes, ils ne peuvent compter qu'en binaire. Mais pour expliquer tout ça, il faut en revenir à comment les humains comptent.

## p1: Représenter les nombres (et compter avec)

Je pense que vous savez tous compter sur vos doigts (montrer).

[Anecdote](#): anglophones et chinois ne font pas comme nous: pouce en dernier.

Et pour les Japonais 0=main pleine, 5=main vide.

On compte comme ça depuis la préhistoire (imaginer des usages): c'est la même chose que faire des [encoches](#) sur un os ou un bois de renne. Nos ancêtres ont aussi compté avec des [cailloux](#).

Anecdote: d'ailleurs cailloux se dit *calculus* en latin → calcul, au sens math et au sens calculs rénaux.

Quand on y pense, ça revient à compter 0,1,11,111,1111... (montrer sur les doigts): ça n'est pas un usage très parcimonieux des chiffres, vous savez faire mieux pour compter !

Du coup on ne peut compter que jusqu'à 5 avec une main, 10 avec 2 mains. Si on veut aller plus loin, il faut utiliser toute la famille. C'est un gros gaspillage de doigts. 😊

Et pour faire une addition, il faut égrener le second nombre vers la main qui stocke le premier nombre.

A noter que d'autres civilisations sont plus malines, comme en Egypte, en Asie du sud, et peut-être les Celtes: eux, ils savent compter jusqu'à 12 sur une seule main, avec les phalanges (montrer). Et ensuite, ils sont malins une 2ème fois: ils marquent de la même façon "une douzaine" sur la phalange de la main droite. (puis montrer 13,14...24). et du coup avec les 2 mains, jusqu'à 144 ! (flasher les mains).

Ensuite - preuve que les plus forts ne sont pas forcément les plus malins - , les romains ont utilisé ce machin: vous avez déjà vu les chiffres romains, vous voyez comment ça marche ? (expliquer vite fait: I II III ... IV (=5-1) V VI (=5+1) ... puis autres lettres X,L,C,D,M (et au delà, on fait quoi ?). Vous savez comment faire une addition avec ça ? Moi pas. Et des multiplications ? En fait je doute fort qu'ils faisaient du calcul avec ce truc infâme. Mon hypothèse c'est que ce sont leurs esclaves phéniciens ou autres qui s'occupaient des comptes et des courses, avec un vrai système de numérotation à eux qui marche. 😊

Ensuite, il y a eu l'idée géniale de la notation positionnelle, correspondant aux "chiffres arabes" qu'on utilise aujourd'hui. Ils nous sont arrivés tard, au moyen-âge, via l'Andalousie, creuset de la rencontre entre monde arabe - la civilisation avancée à l'époque, avec les savants et mathématiciens - et le monde médiéval occidental. Mais eux-même n'ont fait qu'améliorer un système qui venait d'Inde, et on pourrait remonter encore avant.

Le principe, c'est d'écrire des chiffres bien alignés en colonnes - on a jamais fait mieux que le tableur pour bien faire les additions 😊.

Anecdote: Les Incas utilisaient déjà qqchose de proche il y a 4500 ans: les quipu, où chaque cordelette verticale code un nombre de 0 à 999, avec 3 zones correspondent aux 3 chiffres, notés en faisant de zéro à 9 nœuds. On a retrouvé comme ça plein de relevés de nombres - peut-être genre production agricole mois par mois ? Ils faisaient même parfois des "dossiers", où les fils portant les nœuds sont à un 3ème niveau, attachés à la cordelette numéro 2 verticale. Et aussi parfois une "étiquette" sur le côté, qui code aussi un nombre. Dommage qu'ils n'aient pas aussi utilisé un alphabet pour nous dire de quoi il s'agissait !

## p2:

Avec ce système, on se donne un petit nombre de symboles, les *chiffres*.

En base 10, ce sont les chiffres 0,1,...9.

On utilise ça tous les jours sans y réfléchir, mais si on décrit comment ça marche:

428 (montrer), ça veut dire 8 unités, et pour ce qui est + grand on passe à la colonne suivante = 2 dizaines, puis 4 centaines, c'est à dire 4 dizaines de dizaines. Avec les doigts pour 428 il aurait fallu mobiliser tout le village, et là, on a juste 3 chiffres 😊. Et comme ça on peut représenter n'importe quel nombre jusqu'à l'infini avec juste 10 symboles, pas comme les romains.

Et pour faire une addition (montrer), quand on a le chiffre maxi, 9, et qu'on veut lui ajouter 1, alors on doit repartir de zéro, et on a une retenue à ajouter à la colonne d'après, les dizaines.

Si on comptait en base 5, avec que les symboles 0,1,2,3,4... (expliquer la même chose).

(Zapper en version courte). Les Babyloniens, eux, il y a 4000 ans, ils utilisaient la base 60. On leur doit d'ailleurs quelque chose qu'on utilise encore? Vous devinez quoi ? Où on compte jusqu'à 60 ? Oui, pour compter le temps, et les angles: on compte les secondes jusqu'à 60 - enfin, 59 - , et après on ajoute 1 aux minutes et on repart à zéro pour les secondes. Et après 59 minutes, 1h. Mais pour représenter les nombres de 1 à 60, les babyloniens n'avaient pas nos chiffres... et ils n'allaient pas inventer 60 symboles différents. En fait ils ont un truc intermédiaire: ils comptent de 1 à 9 poinçons, et après, un chevron vaut 10 - c'est pas si différent de nous pour les secondes.

Anecdote: pour écrire en cunéiforme sur tablette d'argile fraîche, on utilise un stylet qu'on appuie de différentes façons. → (décodage de 9018).

Mais bon, c'est quand même plus simple avec les bases petites. Et la plus simple de toutes, c'est la base 2, le *binaire*, où on n'a que deux symboles: 0 et 1. Plus simple c'est pas possible: avec que le 0 on pourrait juste compter 0, 00, 000, ça revient à compter sur les doigts.

Et du coup c'est facile à mécaniser: "1" ça peut se dire "ya du courant" et 0 "ya pas de courant".

Ou ya de l'eau / ya pas d'eau. Ou ya une bille / ya pas de bille.

→ Re-expliquer le décodage d'un nombre (\*) et l'addition. (\*): En binaire on appelle les chiffres des *bits*. (NB: certaines classes d'ados peuvent se gausser à \*chaque\* usage de terme "bits" → jager s'il faut éviter ;-))

Mais on peut aussi compter en binaire sur les doigts ! Avant on faisait (re-montrer 1...5 sur les doigts).

Mais là si on dit que pouce=1, index=2, annulaire=4 (en parallèle du décodage), alors on peut décomposer un nombre en addition, par ex 5=1+4. Et du coup ( compter vite juste qu'à 7, pause, puis jusqu'à 15 "déso pour les signes malpolis ;-)" ). Avec 1 seule main, je peux compter jusqu'à 32 (enfin, 32 nombres, de 0 à 31). et avec 2 mains, jusqu'à 1024 !!! (flasher les 2 mains).

→ Vas-y, toi, fais moi un 6... un 7... un 5... ok, vous avez compris !

### p3: Logique Booléenne

Donc maintenant on a le binaire, mais l'ordinateur il est assez bête, faire directement des additions, c'est encore trop compliqué pour lui. Alors pour lui faciliter la tâche, on va commencer par des opérations encore plus simples: les opérations ET, OU, NON.

Ici, 0 veut dire "faux", et 1 veut dire "vrai".

L'opération logique "a ET b", elle veut dire "vrai si a et b sont vrais en même temps". Pour OU, il faut qu'au moins un des deux soit vrai. Mais attention, le français, parfois c'est ambigu: il y a 2 sortes de "ou". Quand on vous demande "est-ce que tu veux une pomme ou une banane ?", si vous répondez "oui" vous aurez l'air bête même si c'est vrai 😊. En fait dans cette question le "ou" attendu c'est le *ou exclusif*: "l'un ou l'autre, mais pas les deux". Et le NON, ça veut juste dire "contraire de". (→ détailler la table de vérité des opérateurs.)

Option: En vrai on a juste besoin de définir la table de 2 de ces opérateurs, et on peut en déduire les deux autres. Par exemple le ou exclusif, "l'un ou l'autre mais pas les deux", peut se traduire en équation logique:  $(A \text{ ou } B) \text{ et non}(A \text{ et } B)$ . De même pour Et: le contraire de "A et B est vrai" c'est "A est faux ou B est faux", soit  $\text{non}(A) \text{ ou } \text{non}(B)$ . Du coup,  $A \text{ Et } B = \text{non}(\text{non}(A) \text{ ou } \text{non}(B))$ .

A partir de là, on aime bien faire des raisonnements logiques avec des petits dessins: chaque opérateur logique est dessiné par un symbole, une "[porte logique](#)" (montrer), prenant en entrée 2 fils correspondant aux paramètres A et B (qui valent vrai ou faux, 1 ou 0) et en sortie un fil avec le résultat. Option: ici on a un long tunnel théorique. Il manquerait ici une activité pour réveiller. Briques lumineuses pour faire le ou exclusif (ou le et) à partir des autres portes ? Ou des questions-quiz malines à la salle ?

Maintenant ce qu'on aurait voulu, c'est une porte "+", qui fait l'addition des deux bits entrées, et qui sort d'un côté la somme, de l'autre la retenue (re-montrer la table d'addition).

Si on regarde à nouveau la table d'addition (la re-détailler), la somme c'est exactement le ou exclusif (comparer la colonne avec la table du OuX), et c'est normal puisque ça vaut 1 si A ou B vaut 1 mais pas les deux. Et il y a une retenue uniquement si A et B valent 1, du coup ça donne exactement le Et (comparer la colonne avec la table du Et).

Donc une addition, ça correspond à ce schéma (re-détailler ce qu'il s'y passe).

Et là, on est pratiquement arrivé au point le plus difficile de l'exposé; à partir de maintenant on va juste faire du Lego avec ça.

A ce stade il y a quand même une petite arnaque: là, on sait juste ajouter deux bits; deux chiffres 0 ou 1 isolés, mais pas deux nombres complets. Comment faire pour un nombre à plusieurs bits ? La seule différence, par ex si on en est à ajouter les 2èmes bits de A et B (montrer l'addition), c'est qu'on doit peut-être prendre en compte une retenue qui vient de l'addition du 1er chiffre. Il faut alors aussi l'ajouter. Mais c'est juste une addition de plus, et on vient de voir comment faire les additions → Pour faire la somme complète de deux chiffres, il faut donc enquiller 2 blocs "addition". Et pour avoir la somme complète de 2 nombres A et B à plusieurs bits (montrer), on cascade le schéma autant de fois qu'il faut.

ça a l'air compliqué, mais comme souvent en informatique, on a défini une bonne fois le petit truc complexe, et ensuite on le duplique comme du Lego, plus besoin de réfléchir au détail.

( récap: re-montrer où on met les bits de A et B, et comment on récupère S - qui peut long d'un bit de plus ).

Notez que jusque là, on a juste fait une sorte de circuit électronique en papier, pour détailler un fonctionnement logique. Maintenant pour le fabriquer en vrai, à vous d'inventer comment faire le 0/1 avec de l'électricité, de l'eau, des billes, etc, puis de trouver comment construire des portes logiques ET, OU, OUexclusif, NON (au moins deux, et on déduit les 2 autres ) avec ça. Et à ce moment, vous aurez une vraie machine à additionner qui marche !

#### p4: Comment fabriquer une machine qui réalise ça... en électronique

Avec de l'électricité, on va dire 1 = ya du courant, 0 = ya pas de courant.

Pour faire un Ou, l'idée naïve ce serait juste de relier les fils de A et B: ça marche, mais le problème c'est que si on fait ça en cascade comme sur le schéma, on va se retrouver avec bc trop de courant à la fin. En plus, si A et B sont à 1 tous les 2, en C on aura une sorte de 2, et pas 1, ça peut fausser la suite.

→ La solution est de faire ça de manière plus indirecte: s'il y a du courant en A, on va *déclencher* C, et idem pour B.

Pour déclencher, en électronique on a un composant exprès pour, le *transistor*, qui est une sorte d'interrupteur qui laisse passer le courant entre l'entrée (la patte de gauche) et la sortie (la patte de droite) s'il y a du courant dans la patte du milieu.

( au moins pour les plus jeunes on zappe les autres explications du transistor, et on y reviendra un peu plus bas ).

Vous avez déjà fait des travaux manuels avec une pile et une ampoule, et des fils et un interrupteur ? Voici comment on ferait notre OU avec ça (montrer le schéma):

Une borne de la pile arrive en haut (1V), et l'autre en bas (0V = la masse).

On veut qu'une lampe s'allume en C si on appuie sur l'interrupteur A ou l'interrupteur B. Pour ça on les fait relier C à la pile indépendamment, "en parallèle". On voit bien que si on appuie sur A ou sur B, le courant va passer. Et si on n'appuie sur rien, l'ampoule est reliée au zéro volt.

Option: notez qu'on a mis une résistance entre C et le zéro volt. Une résistance, ça freine au maximum le courant qui veut passer. Pourquoi on fait ça ? Si y'avait pas la résistance, quand on appuie sur B, le courant pourrait aller directement du 1V au 0V, ça fait un court-circuit, pas une bonne idée. En plus C serait relié à la fois au 1V et au 0V, on ne sait pas trop ce qu'il vaudrait.

Option: notez que la résistance peut-être énorme, on n'a aucun besoin de consommer du courant: ce qui fait que C vaut 1 ou 0, c'est juste qu'il soit relié au 1V ou au 0V, pas besoin que du courant passe vraiment.

Et pour le ET: (même histoire, mais en série).

(retour aux transistors)

Là on a montré avec un interrupteur et une ampoule, mais dans un circuit électronique c'est exactement pareil avec un transistor, où c'est le courant (= 1) qui arrive en A ou B qui déclenche. Un transistor, si vous avez déjà cassé un poste de radio, ça ressemble à ça (composants à 3 pattes). Dans le poste à lampe de vos arrières-grands parents qui écoutaient "ici Londres", ça ressemblait à ça ([ampoule](#)). Mais de nos jours dans un circuit intégré ça ressemble à ça, qui fait moins d'un micron de large ([au milieu](#)). Et dans un circuit intégré d'1 cm<sup>2</sup>, il peut y en avoir des milliards ([gauche](#)).

ça y est, on a défini comment faire 0 et 1 (courant/pas courant), on a trouvé comment faire des portes logique ET et OU. Donc on pourrait réaliser une machine à additionner électronique, il suffit de refaire le grand schéma !

#### p5: Comment fabriquer une machine qui réalise ça... avec de l'eau

Avec de l'eau, on va dire que 1 c'est "il y a de l'eau qui passe" et 0 c'est "il n'y a pas d'eau".

Regardez ce [composant](#) (montrer): il y a deux entrées d'eau A et B en haut. Si de l'eau arrive en A, elle va se cogner au côté droit du creux puis sortir en bas en S. Idem si l'eau arrive de B.

Mais s'il y a de l'eau qui arrive des 2 côtés en même temps, alors les jets se cognent et tombent dans le godet du milieu, qui sort en R. Bref s'il y a de l'eau en A ou B mais pas les deux ça sort en

S: c'est exactement un Ou exclusif, qui correspond au résultat de l'addition. Et s'il y a de l'eau en A et B en même temps alors ça sort en R; c'est exactement un ET, qui correspond à la retenue. → ce composant à lui seul réalise un (demi)-additionneur complet ! (par défaut on n'a pas posé le terme de demi-additionneur sur le 1er schéma sans retenu. à introduire, optionnellement).

En vrai, il a quand même deux défauts:

- en R il va sortir 2 fois trop d'eau, il faut en jeter la moitié (pas bien dur à faire: par ex, petit trou).
- mais surtout pour que le ET marche, il faut que l'eau arrive exactement en même temps en A et B, sinon ça coule à côté (dans S) et ça peut calculer faux. Mais on a vu que la retenue va mettre du temps à se propager d'un chiffre à l'autre, alors ça ne va pas bien marcher.

→ Nous on se sert d'un [autre principe](#): on va utiliser un siphon, un peu comme dans vos WC (\*) ! Vous avez déjà vu comment marche un siphon ? (en général non → démonstration avec un tuyau transparent type aquarium (au pire, une paille) entre un verre haut et un verre bas). Si de l'eau peut arriver + haut que le point haut du tuyau (aspirer), alors ça va se mettre à circuler tout seul sans s'arrêter, jusqu'à vider tout le verre, sans électricité ! Ici on a aspiré, mais il suffirait que ça se remplisse tout seul et si ça dépasse le niveau max ça déclenche. Magique, non ?

(\*) Détail optionnel, si question: En vrai les siphons de WC ne marchent pas toujours par "effet siphon". Uniquement ceux à sortie verticale.

Dans notre cuvette, on fait se verser les entrées d'eau A et B. S'il n'y a de l'eau qui vient que d'un côté, alors la cuvette est à moitié remplie: on fait sortir ça doucement par un tuyau fin pour comptabiliser le résultat S. Mais s'il y a de l'eau qui arrive à la fois dans A et B, le siphon va se déclencher - même si l'eau n'arrive pas pile en même temps - et vider toute la cuvette. Ce qui donne la retenue R, et il reste 0. Et en S il sortira de l'eau s'il y avait un jet en A ou en B mais pas les deux: Ou exclusif.

Bref ici aussi on a un composant qui fait tout en un: le siphon fait la retenue R, et ce qui reste fait la somme S.

A partir de là, plus qu'à en enquiller deux pour réaliser une somme de 2 bits avec retenue, puis en cascader plusieurs pour faire la somme de plusieurs chiffres/bits comme vu avec le grand schéma (montrer), et on a un additionneur ! Et c'est exactement ce qu'on a réalisé en impression 3D (montrer). Et que vous allez expérimenter maintenant pour vérifier si ça marche. Allez, approchez-vous !

→ manip sur les additionneurs en [impression 3D](#), + seringues mouche-bébé, + eau avec colorant alimentaire (prévoir éponge/sopalin/ voire serpillère ;-)). On peut prévoir plusieurs spots avec additionneur, seringue, cuvette, éponge.

Montrer la manip: Mettre les additionneurs à plat sur la table, choisir 2 nombres A et B à additionner, en binaire (plus intéressant si retenues).

Seringuer les bits de biais vers le centre des cases à remplir aux  $\frac{3}{4}$ , puis mettre verticalement pour le calcul. On ne triche pas: vérifier qu'ils savent quel est le nombre décimal encodé, pour vérifier le résultat.

Ensuite redresser assez vite bien verticalement pour lancer le calcul, et suivre le jeu de l'eau. Notamment les derniers siphons, qui se remplissent patiemment avant de déclencher leur retenue (si tout se passe bien. Suspense... 😊).

Bien secouer pour vider les bulles avant réusage.

Anecdote: ce principe a vraiment été envisagé pour réaliser une [fontaine-additionneuse](#) dans un parc de Londres, où les promeneurs remplissent les bits de A et B avec des robinets.

Et un artiste a réalisé une [horloge](#) numérique à eau sur ce principe.

## p6: Comment fabriquer une machine qui réalise ça... avec n'importe quoi

Comme on l'a vu il suffit de choisir ce qui représentera physiquement 0 ou 1, puis ensuite de trouver comment réaliser des portes logiques. Il suffit d'avoir de l'imagination, de la patience, et pas mal de méticulosité 😊.

Des gens l'ont fait avec des quantité de choses différentes:

- (zapable) Il existe des additionneurs avec des engrenages (bon ok, en vrai c'est rarement du binaire). Par exemple les caisses enregistreuses dans les westerns - ya pas d'électricité à l'époque ! Les premières ont été inventées dès le XVIIème siècle, comme la [Pascaline](#).
- En hydraulique (zapable, notamment pour les jeunes qui ne savent pas encore conduire): En France on utilise des boîtes de vitesse mécanique, mais aux Etats Unis ils utilisent essentiellement des boîtes automatiques, et ça depuis longtemps. Aujourd'hui c'est tout électronique, mais à l'époque il fallait bien que quelque chose d'autre calcule s'il faut changer de vitesse, en fonction de la pédale d'accélérateur et du régime moteur, et l'effectue concrètement. Hé bien c'est un [réseau hydraulique](#), dans ce labyrinthe, qui va le faire: la pression d'huile venue des entrées (pédale d'accélérateur, régime moteur) va s'affronter le long de portes, et au final déclencher des valves qui vont débrayer, changer d'engrenage et ré-embrayer !
- Par ondes hyperfréquences (on pourra zapper selon âge et expérience): On peut faire des [circuits](#) directement avec les ondes venues de votre antenne télé, ou d'un radar, en les piégeant dans des pistes de cuivre ou même juste dans des barres métalliques creuses ! Par ex dans un bras en T une onde va partir et rebondir, et causer une interférence qui peut annuler le signal; on peut retarder une onde avec un zigzag, etc. Bref, on a une sorte de dessin qui va directement agir sur les ondes et leur faire faire des opérations du même genre qu'on a vu.

## p7 (suite)

- Biologie (zapable) : On sait faire des [manips](#) avec des brins d'ARN ou d'ADN qui vont aussi réaliser des opérations logiques de base. Mais il faut que quelqu'un transvase les produits d'un récipient à l'autre pour les diverses étapes; c'est un peu de la triche 😊.
- Par contre dans nos neurones (NB: se voit en 3ème) il se passe des choses plus proches: Le corps cellulaire d'un neurone est muni de [dendrites](#) où viennent se brancher les copains. Si le neurone se déclenche, l'influ voyage le long de son axone, jusqu'à ses [synapses](#) tout ou bout qui vont titiller ses autres copains. Si le seuil de déclenchement du neurone est réglé très bas, il suffit qu'une des synapses titille le neurone pour que ça le déclenche: c'est comme un OU. Par contre si le seuil du neurone est réglé très haut, il faudra que toutes les synapses soient actives pour qu'il se déclenche, c'est comme un ET. Bon en vrai un neurone n'est pas une simple porte logique (il peut y avoir des synapses inhibitrices, des qui court-circuitent les dendrites en se connectant directement sur le corps cellulaire, etc), mais ça garde le même principe: quelque chose de très simple connecté des milliards de fois, et au final ça peut donner qqchose d'aussi complexe que vous qui rêvez à votre prochain WE ! 😊

## p8 (suite)

- Avec des dominos: c'est assez facile d'imaginer un OU: il suffit que la chaîne de dominos chute à l'entrée gauche ou droite ([gauche](#)). Par contre pour faire un ET c'est plus compliqué. Un demi-additionneur, ça va donner un schéma comme ça ([milieu](#)). Et des gens ont vraiment réalisé un additionneur complet de nombres à plusieurs bits dans un stade, comme ici (droite). Par contre bien sûr ça ne fonctionne qu'une fois. Ensuite pour recommencer un autre calcul il faut relever les dominos à la main (mais on pourrait imaginer un système avec des ficelles pour les relever).
- En [origami](#) (zapable): un [pliage spécial](#) fait que le pli de sortie ne se couche que si l'on appuie sur un des deux plis du haut, et un autre le fait seulement quand on appuie sur les deux en même temps. Du coup on peut réaliser un grand pliage additionneur: au départ on marque les plis correspondant aux bits à 1, puis on passe la main le long du pliage pour propager le couchage ou non des plis (j'ai essayé, en vrai c'est pas évident à faire marcher).
- En simulation sur ordi: il existe un logiciel un peu méconnu: ça s'appelle Minecraft, vous avez entendu parler ? 😊 ( sursaut puis "Ouiiiii!" chez 96% des jeunes ). Et la [Redstone](#) dans Minecraft, vous avez utilisé ? ( 30 à 70% de oui selon niveau ). Le principe, c'est que ce sont des briques qui créent ou propagent l'énergie, comme des fils électriques. Si 2 fils se rejoignent ça nous fera un OU. Et il y a des briques spéciales, comme la torche: s'il n'y a pas d'énergie à l'entrée elle s'allume, mais s'il y a de l'énergie qui arrive elle s'éteint → ça fait une porte logique NON. Hop, on a 2 portes; et comme on l'a vu, avec 2 portes on peut faire les 2 autres, ici "Et" et "Ou exclusif", et donc tout ce qu'il faut pour faire un additionneur. Des gens l'ont fait ([gauche](#)), et même avec un écran de visualisation (droite). En fait le + gros du boulot, c'est d'allumer les pixels (faits avec une brique lampe) correspondant à chaque chiffre possible 😊. Mais ya des fous qui ont fait bien pire: Un vrai [ChatGPT](#) qui marche ! très lent, avec pas beaucoup de mémoire, mais qui marche ! En utilisant 440 millions de briques (il ne les a pas mises à la main, il a fait un programme pour ça 😊).

D'autre encore on fait des additionneurs avec des billes: 1 = ya une bille, zéro = pas de bille, deux qui arrivent et ça fait une bascule (comme pour le siphon)...

Pour conclure, comme on l'a vu pour faire un système qui calcule il suffit de choisir ce qui représentera 0 ou 1 avec le matos que vous voulez, puis ensuite de trouver comment réaliser 2 portes logiques. Et en empilant ce qu'il faut, on peut réaliser n'importe quoi: une calculatrice, un ordinateur complet, un jeu comme Doom, ou ChatGPT. Il suffit d'avoir de l'imagination, pas mal de patience, et aussi beaucoup de méticulosité 😊. Mais ça marche exactement comme ça aussi dans votre ordinateur ou téléphone, avec de la bête électricité et des transistors !