SUPPLEMENTARY DATA

Availability of synthetic genetic diversity

Promoters are one of the pivotal elements to engineer regulatory circuits with functional behavior. Promoters could be evolved by random mutagenesis process, followed by screening to select the sequences of interest [S1]. The pool of new promoters can be characterized by using a suitable mathematical model, with kinetic parameters fitted from experimental data. Constitutive promoter libraries have been also constructed from bacterial [S2] and phage (T7) [S3] promoters. Examples of synthetic repressible/inducible promoters are based on well-studied systems such as phages, in particular the mutated λ_R promoter [S4] or the T7 promoter carrying the *lac* operator (lacO) [S5]. Moreover, Collins group has recently constructed a library of *tet* and *lac* promoters in yeast [S6]. After characterizing these regulatory elements, they combined them to design functional and predictable circuits.

Combinatorial promoters allow the cell to perform computations in multifaceted environments to adopt a given gene profile [S7]. For Synthetic Biology, such promoters can integrate different signals allowing the design of complex signal-processing circuits. Although in Nature there are a lot of examples of promoters controlled by many regulators, synthetic applications have used no more than three transcription factors. In the past, Joung and coworkers illustrated the engineering of combinatorial promoters by placing together different operators, in particular crpO and λ_{RM}O [S8]. More of a decade after, many combinatorial promoters have been designed and even integrated in functional circuits, such as the integration of λ_RO and luxO (to construct a pulse-generating circuit) [S9], lacO and luxO [S10], lacO and λ_{RM}O [S11], lacO and araO (to construct a fast oscillator) [S12], and lacO-modified glnA promoter (to construct an oscillatory-bistable circuit) [S13]. At more large scale,

Elowitz and coworkers developed a library of combinatorial promoters based on LacI, TetR, AraC and LuxR to work in bacteria [S14], whereas Collins group reengineered the GAL1 yeast promoter with different tetO at different positions [S15].

Another strategy to generate genetic diversity consists in mutating the binding domain of the transcription factor. Recently, it has been engineered a library of *tet* repressor mutants, which have specificity for a synthetic operator containing four base-pair replacements [S16]. Importantly, the mutant collection is orthogonal to the wild-type TetR. Moreover, zinc-finger proteins offer the possibility of engineering specific regulatory elements working in several organisms, taking advantage of the modular composition of the helixes and the specific DNA-binding domains [S17].

Furthermore, post-transcriptional regulatory elements can control protein expression, thus adding more richness to the design. There are different mechanisms by which small non-coding RNAs that bind to their mRNA target perform regulation at this level. The design principles from natural examples are exploited to reengineer synthetic systems. For instance, chemical compounds or small *trans*-RNAs can trigger riboswitches [S18,S19] or repress tRNAs to avoid stop codons [S20]. Importantly, these systems can be used in series together with transcription control elements. On the other hand, in addition to libraries of ribosome binding sites, computational framework allows creating a sequence *à la carte* for a specific translation rate of a target protein [S21]. Remarkably, riboregulation at a high transcription rates produces less noise levels then allowing a more precise control of protein expression [S22].

Mathematical expression for transcription rate

The transcription rate *f(y,u)* we use here is based on Hill-type functions [S7,S23]. Accordingly, this term usually reads

$$f(y,u)=\frac{\alpha_0+\alpha\left(\dfrac{y}{\theta\left(1+\left(u/\vartheta\right)^{\sigma}\right)}\right)^{\nu}}{1+\left(\dfrac{y}{\theta\left(1+\left(u/\vartheta\right)^{\sigma}\right)}\right)^{\nu}}=\frac{\alpha_0+\alpha R(y,u)}{1+R(y,u)}, \qquad \text{(S1)}$$

where α_0 and α are the transcription rates of free and bound promoter, respectively. In case of transcriptional activation $\alpha_0<\alpha$, whereas $\alpha_0>\alpha$ for repression. For protein *y*, θ is the regulatory coefficient (it may be viewed as the apparent dissociation constant of protein-DNA) and ν the Hill coefficient (it may be viewed as the degree of multimerization). For cofactor *u*, ϑ and σ are the corresponding regulatory and Hill coefficients, respectively. In addition, using the previous formalism and denoting by *R(y)* the regulatory factor (e.g., $R(y)=(y/\theta)^{\nu}$ in absence of cofactor molecule), we can account for combinatorial regulation [S7]. For illustrative purposes, let us show a function of two regulators

$$f(y_1,y_2)=\frac{\alpha_0+\alpha_1R(y_1)+\alpha_2R(y_2)+\alpha_{12}\omega R(y_1)R(y_2)}{1+R(y_1)+R(y_2)+\omega R(y_1)R(y_2)}, \qquad \text{(S2)}$$

where ω is the cooperation coefficient. For competitive binding $\omega<1$ and for cooperative binding $\omega>1$; $\omega=1$ corresponds to independent binding. It is important to point out that the equations (S1) and (S2) are derived from single kinetic reactions by assuming a quasi-steady state (i.e., binding reactions much faster than transcription) and neglecting the amount of protein bound to DNA with respect to the one that is free.

Stochastic simulation via Langevin model

To perform stochastic simulations, we use a Langevin formulation accounting for intrinsic and extrinsic noise [S24,S25]. In that case, the model reads

$$\frac{dx_i}{dt} = Cf(y_j,u_j) - (\delta+\mu)x_i + \sqrt{Cf(y_j,u_j) + (\delta+\mu)x_i}\,\xi_{x_i}(t) + q_g\xi_g(t) \quad \text{(S3)}$$

and

$$\frac{dy_i}{dt} = g(x_i,x_j) - (\beta+\mu)y_i + \sqrt{g(x_i,x_j) + (\beta+\mu)y_i}\,\xi_{y_i}(t) + q_g\xi_g(t) \quad \text{(S4)}$$

where $\xi_i(t)$ are Wiener processes, with statistics $\langle \xi_i(t) \rangle = 0$ and $\langle \xi_i(t)\xi_i(t') \rangle = \delta(t'-t)$ for the intrinsic noise, and $\langle \xi_g(t) \rangle = 0$ and $\langle \xi_g(t)\xi_g(t') \rangle = \frac{\mu}{2} e^{-\mu|t'-t|}$ for extrinsic noise, because it is exponentially distributed and its autocorrelation time is given by the cell doubling time (μ represents growth rate) [S26]. The parameter q_g gives the amplitude for that noise, and in case of only considering intrinsic noise we set q_g=0. To solve these equations, we consider the noise term constant in one time interval [S27]. Then, for a generic system $dx/dt = P(x) + Q(x)\xi(t) + q_g\xi_g(t)$, and given z_1 and z_2 two Gaussian standard normally distributed random numbers, the solution reads

$$x(t+\Delta t) = x(t) + \int_t^{t+\Delta t} P(x(s))ds + z_1 Q(x(t))\sqrt{\Delta t} + q_g\xi_g(t), \quad \text{(S5)}$$

with

$$\xi_g(t+\Delta t) = e^{-\mu\Delta t}\xi_g(t) + z_2\mu\sqrt{\Delta t}\,. \quad \text{(S6)}$$

Computationally, we solve deterministically the system in one time interval and then apply a stochastic perturbation to update the solution. These values were used as initial conditions to solve the next time interval. Likewise, we solve the stochastic dynamics for the whole time domain.

Post-processing the dynamics of a set of circuits

After simulating a large set of circuits, the dynamics of each for every input condition is stored for post-processing. As opposite to the optimization stage, where a fitness function is computed to *in silico* evolve the circuit, now a Boolean function is applied to indentify a functional circuit. For instance, a circuit that changes its output concentration more than 10-fold according to one external inducer will be considered as a YES gate (using a fold-change threshold of F=10). This allows selecting functional circuits not by their absolute concentrations but by their relative expression levels of operation, thanks to the consideration of such a fold-change threshold. In case of an AND gate (and also extensible to other logic gates), the four entries of the truth table (X_{00}, X_{01}, X_{10} and X_{11}, being X_{ij} the output concentration for input levels *i* and *j*) have to be satisfied by ensuring that $X_{11} > F\ X_{00}$, $X_{11} > F\ X_{01}$, and $X_{11} > F\ X_{10}$. In case of an amplitude filter, we have to ensure that $X_{1/2} > F\ X_0$ and $X_{1/2} > F\ X_1$. In case of an oscillator, the circuit has not to reach a stable steady state. We also considered circuits with damped oscillatory dynamics during a large time.

When the stochasticity of the cell is taken into account, we imposed a further condition to ensure that noise does not affect the functional behavior of the circuit, which for a YES gate turns into $\langle X_1 \rangle > F\langle X_0 \rangle$ and $\langle X_1 \rangle - \Delta X_1 > \langle X_0 \rangle + \Delta X_0$. This can be extended to other behaviors. To distinguish between oscillations and noise in stochastic simulations, we passed to the Fourier space, identifying the existence of oscillations when the mass of the power spectral density is concentrated around a given frequency. For that, we used an algorithm based on the fast Fourier transform, avoiding very high and very low frequencies.

Additional Figures

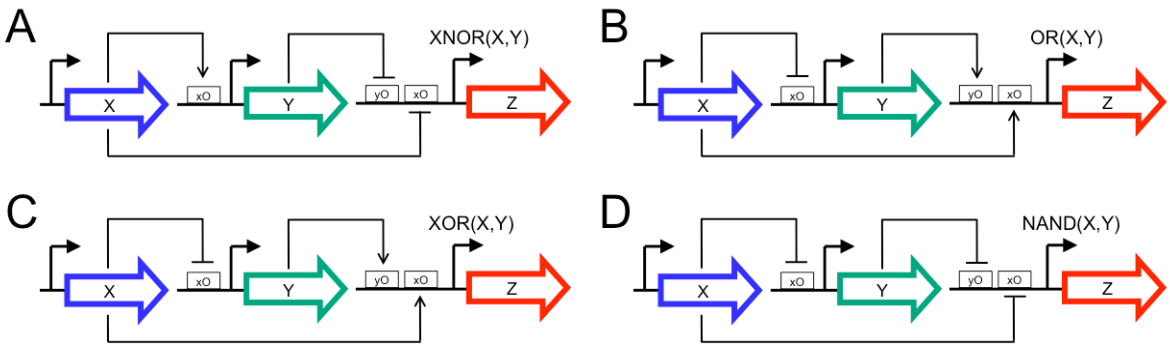

Figure S1: FFL-based gene circuits for working as inverse amplitude filters.

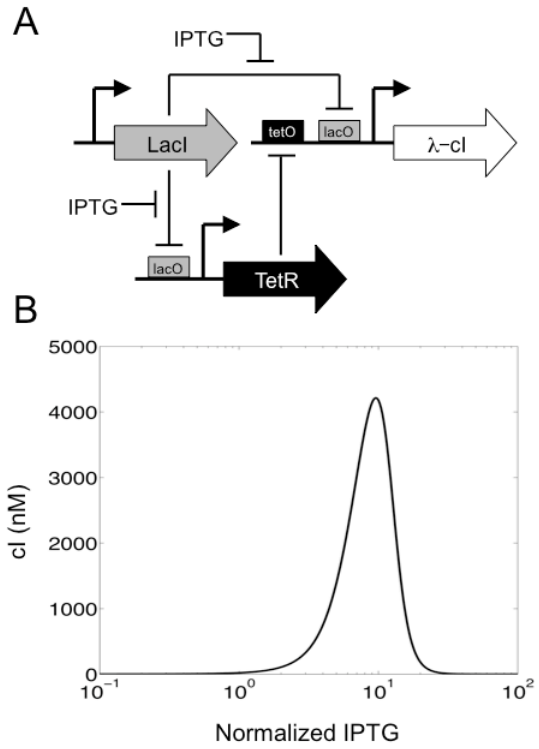

Figure S2: (A) Scheme of a genetic circuit automatically designed to function as an amplitude filter. IPTG is the input of the system and λ-cI is the output gene. The circuit is constructed by using six parts from the library. The mathematical model is provided in SBML format in the Supplementary File sbml.zip. (B) Transfer function of the circuit relating the expression of λ-cI (in steady state) to the concentration of IPTG. Normalized IPTG is the ratio between the concentration of IPTG and its dissociation constant with LacI.

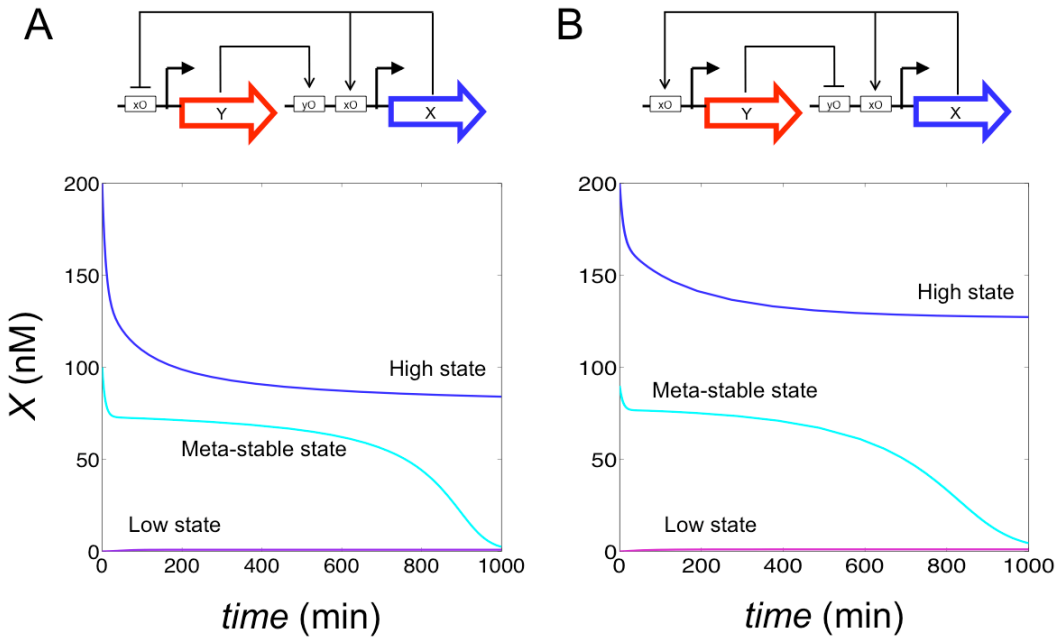

Figure S3: Scheme of two two-gene circuits for working as memory-like devices with an activator-repressor core, together with their dynamics showing bistability and a meta-stable state.

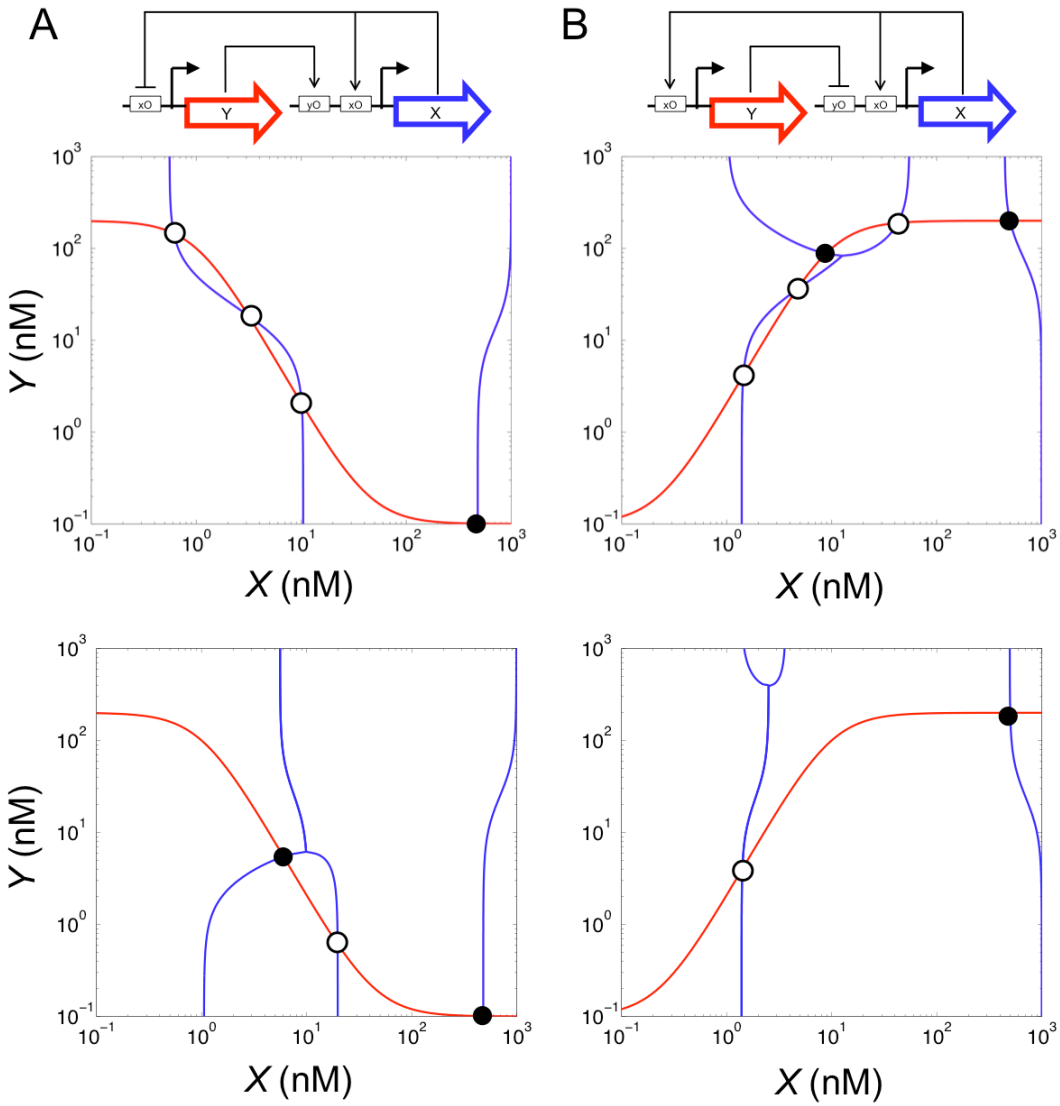

Figure S4: Scheme of two two-gene circuits for working as memory-like devices with an activator-repressor core, together with their phase diagrams for two different parameter sets. Blue lines are the nullclines for gene *X* and red ones for gene *Y*. Filled circles represent stable steady states, whereas open circles unstable states. These two systems, with a proper parameterization, can reach bistability and meta-tristability.

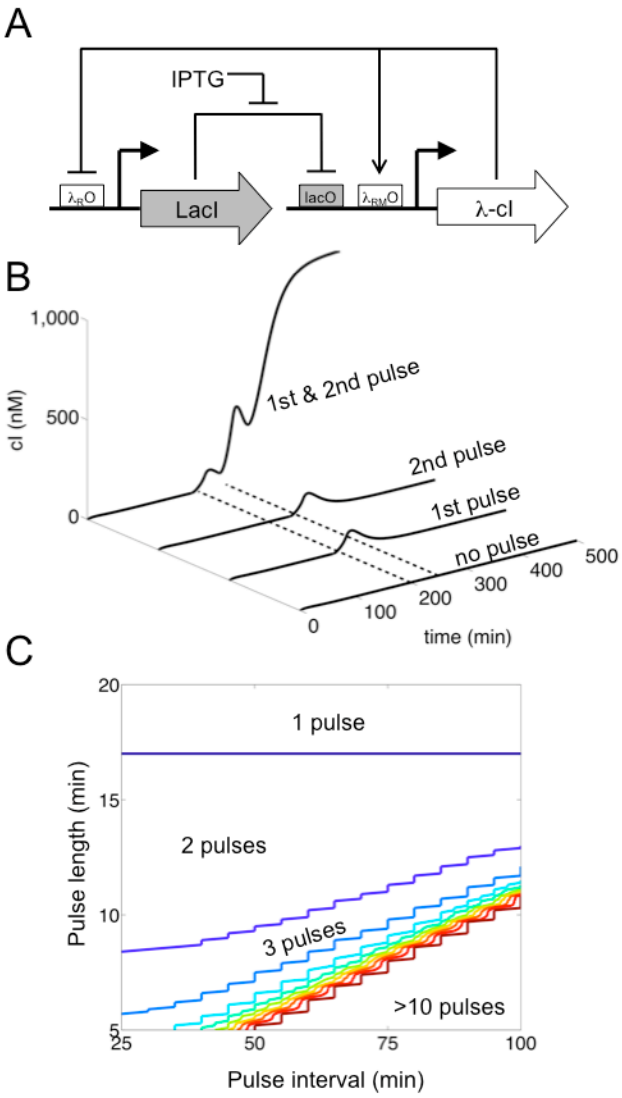

Figure S5: (a) Scheme of a genetic circuit automatically designed to function as a two-pulse counter. IPTG is the input of the system and λ-cI is the output gene. The circuit is constructed by using four parts from the library. The mathematical model is provided in SBML format in the Supplementary File sbml.zip. (b) Dynamical behavior of the circuit responding to none, one (the first or the second) or two pulses of IPTG. The first pulse is applied at 200 minutes, and the second 50 minutes later. As design specification, the pulse length is 10 minutes. Normalized IPTG is the ratio between the concentration of IPTG and its dissociation constant with LacI. (c) Circuit behavior (number of pulses that is able to count) as a function of the pulse length and interval.

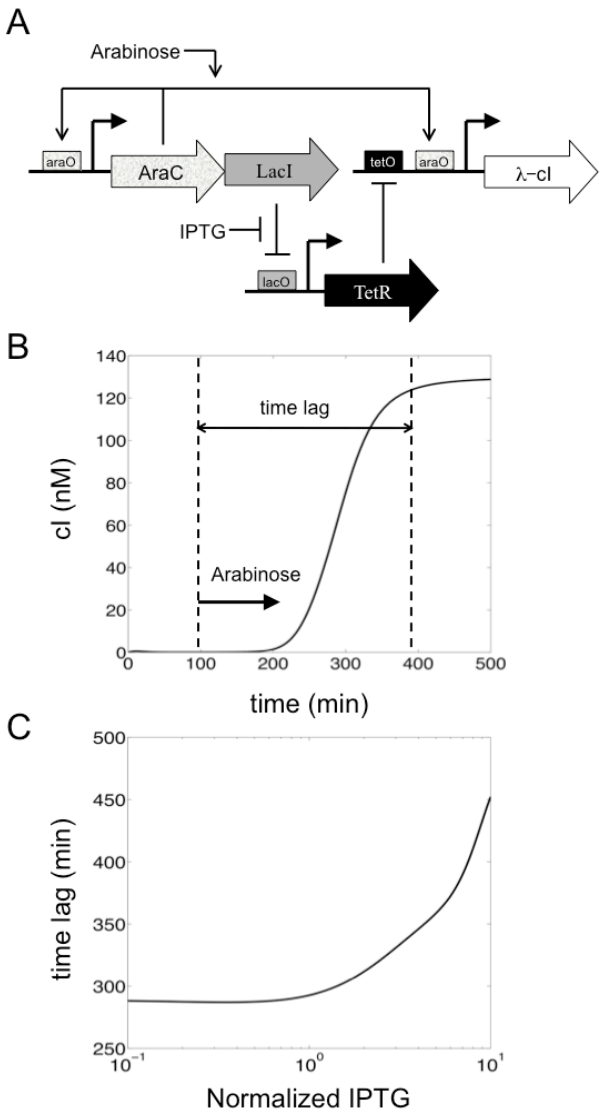

Figure S6: (A) Scheme of a genetic circuit automatically designed to function as a tunable timer. IPTG and Arabinose are the inputs of the system and λ-cI is the output gene. The circuit is constructed by using seven parts from the library. The mathematical model is provided in SBML format in the Supplementary File sbml.zip. (B) Dynamical behavior of the circuit responding to a step of Arabinose applied at 100 minutes without IPTG in the medium. (C) Transfer function of the circuit relating the time lag (time to reach the 95% of the steady state) to the concentration of IPTG. Normalized IPTG is the ratio between the concentration of IPTG and its dissociation constant with LacI.

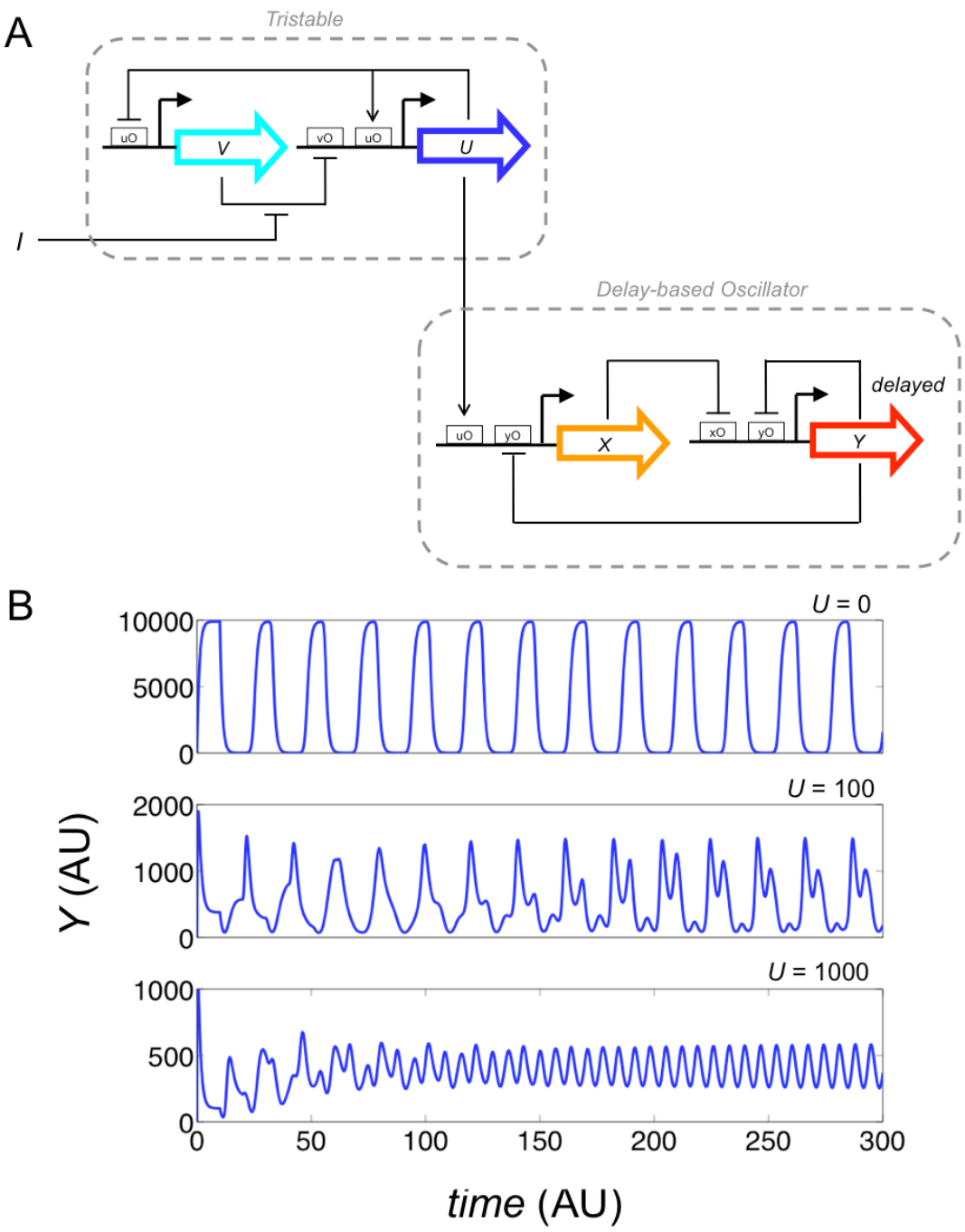

Figure S7: (A) Design of a delay-based oscillator coupled to an upstream module working as a tristable. (B) Output dynamics for each state. AU denotes arbitrary units.

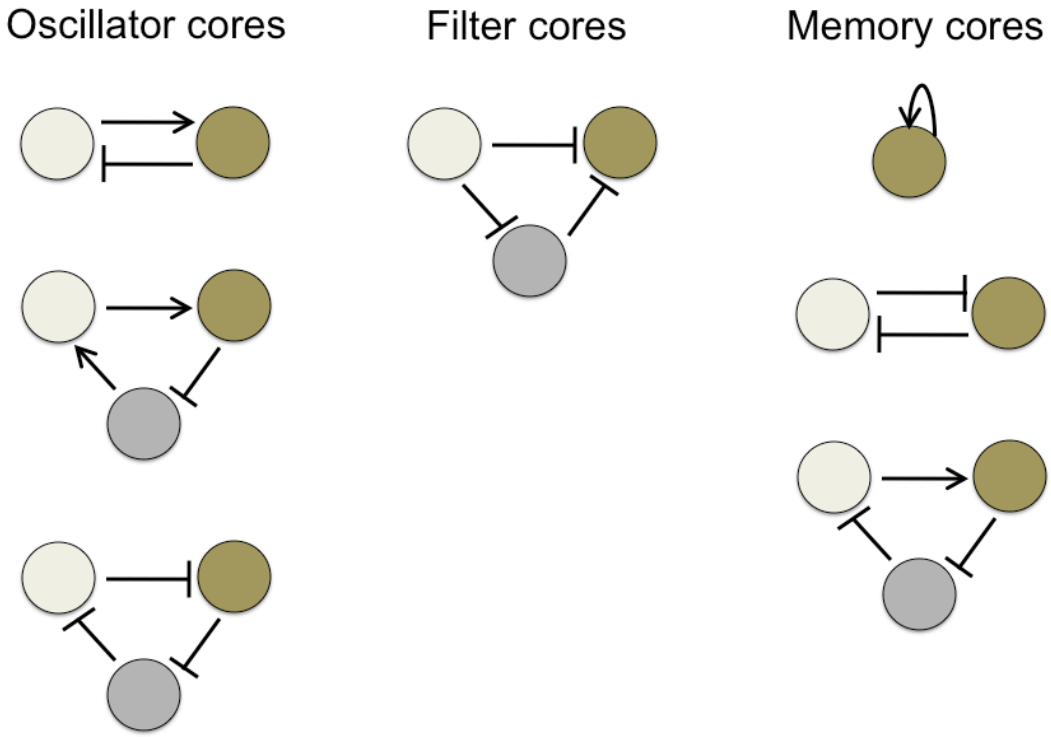

Figure S8: Genetic cores that define the design space of functional circuits provided the library of composable parts (promoters and coding regions) shown in Fig. 5.

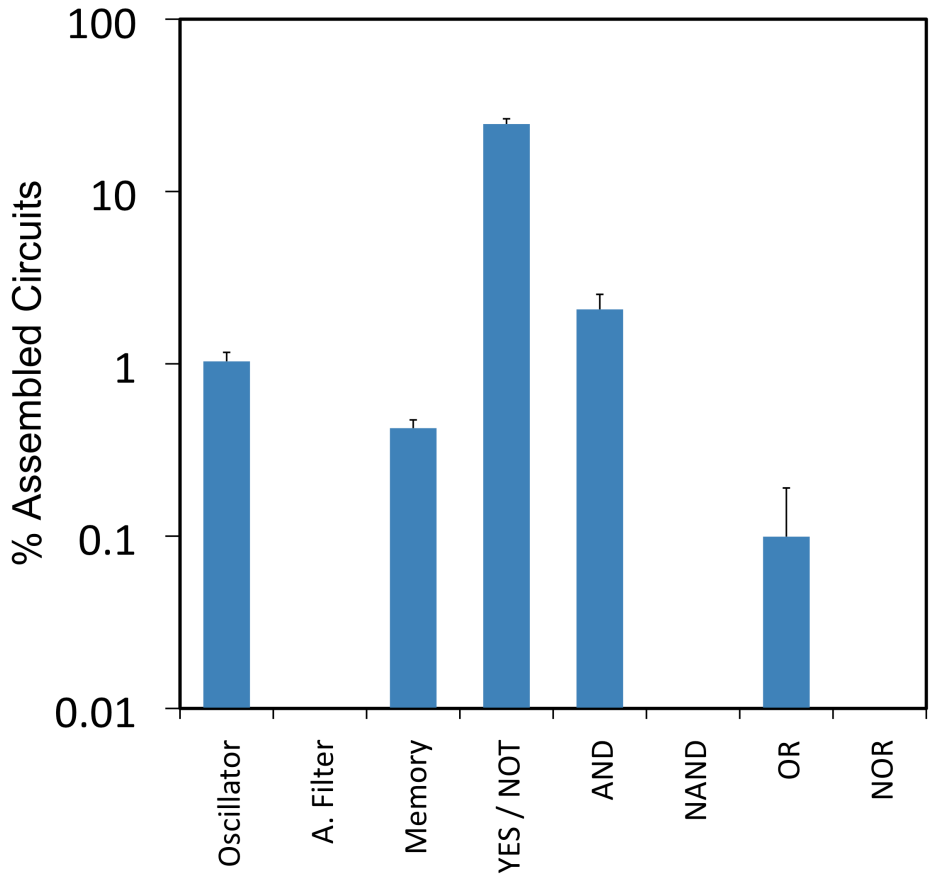

Fig. S9: Dynamical spectrum of the library by exhaustive exploration of different samplings of assembled circuits (about the 0.2% of the circuits). We represent the percentage of circuits that behave as oscillators, amplitude filters, memories, and logic gates (designability). To differentiate between two states of a circuit, we imposed at least one order of magnitude in concentration.

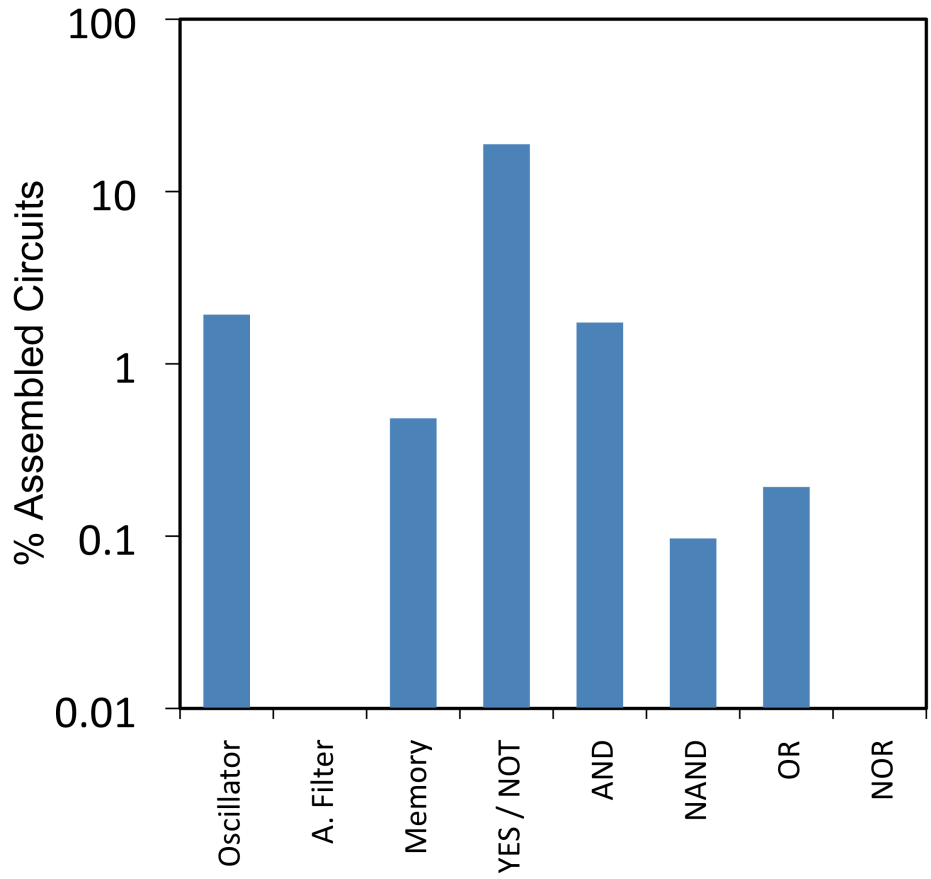

Fig. S10: Dynamical spectrum of the library by exhaustive exploration of one sampling of assembled circuits (about the 0.2% of the circuits) using stochastic simulation. We represent the percentage of circuits that behave as oscillators, amplitude filters, memories, and logic gates (designability). To differentiate between two states of a circuit, we imposed at least one order of magnitude in concentration and avoidance of overlapping in concentration due to the noise.

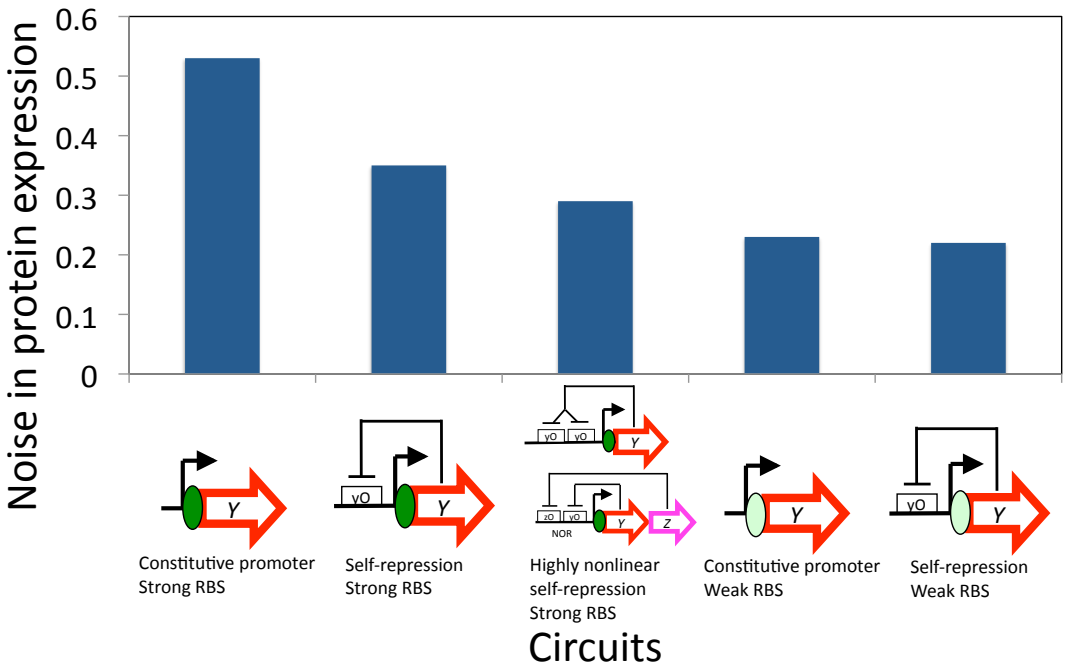

Fig. S11: Design by optimization of circuits that minimize the noise in protein expression (measured as the coefficient of variation of the dynamics, i.e., $\eta_Y = \Delta Y / \langle Y \rangle$). We used stochastic simulations accounting for both intrinsic and extrinsic noise sources (with q_g=1). The mathematical models are provided in SBML format in the Supplementary File sbml.zip. The optimization method also gave a circuit with two independent polycistronic repressors (NOR-like promoter) as a way of reducing the noise in the output by increasing the nonlinearity of the promoter.

Additional Tables

Table S1: Percentage of circuits with a given behavior as a function of the fold-change imposed to differentiate two states (high/low concentration).

F	Oscil.	A. Filter	Memory	YES/NOT	AND	NAND	OR	NOR
1	1.028	0.1576	0.916	41.271	6.441	0.0002	1.455	0.0010
10	1.028	0.0155	0.436	25.172	2.162	0.0002	0.148	0.0010
100	1.028	0	0.374	9.874	0.198	0.0002	0.008	0.0002

Table S2: Number of circuits in common between two functions. In brackets, the corresponding P-values by bootstrapping (1000 replicates).

	Oscillators	A. Filters	Memories	2D Logic Gates
Oscillators	5160	16 ($<10^{-4}$)	36 (0.11)	804 ($<10^{-4}$)
A. Filter	-	78	0 (-)	62 ($<10^{-4}$)
Memories	-	-	2188	351 ($<10^{-4}$)
2D Logic Gates	-	-	-	11099

Table S3: Number of circuits with multiple functions (oscillator, amplitude filter, memory, or 2D logic gate).

	1 function	2 functions	3 functions	4 functions
Circuits	16044	1212	19	0

Additional Files

The mathematical models of the designed circuits shown in Figs. 2, 3, 4, 7, S2, S4, S5, S6, and S11 as well as the library of composable part models shown in Fig. 5, are provided in SBML format in the Supplementary File sbml.zip.

References

[S1] Edelman, G.M., Meech, R., Owens, G.C., and Jones, F.S. (2000) Synthetic promoter elements obtained by nucleotide sequence variation and selection for activity. *Proc. Natl. Acad. Sci. USA*, **97**, 3038-3043.

[S2] Mey, M., Maertens, J., Lequeux, G.J., Soetaert, W.K., and Vandamme, E. (2007) Construction and model-based analysis of a promoter library for *E. coli*: an indispensable tool for metabolic engineering. *BMC Biotechnol.,* **7,** 34.

[S3] Imburgio, D., Rong, M., Ma, K., McAllister, W.T. (2000) Studies of promoter recognition and start site selection by T7 RNA polymerase using a comprehensive collection of promoter variants. *Biochemistry,* **39,** 10419-10430.

[S4] Rosenfeld, N., Young, J.W., Alon, U., Swain, P.S., and Elowitz, M.B. (2005) Gene regulation at the single-cell level. *Science*, **307**, 1962-1965.

[S5] Dubendorff, J.W., and Studier, F.W. (1991) Controlling basal expression in an inducible T7 expression system by blocking the target T7 promoter with lac repressor. *J. Mol. Biol.*, **219**, 45-59.

[S6] Ellis, T., Wang, X., and Collins, J.J. (2009) Diversity-based, model-guided construction of synthetic gene networks with predicted functions. *Nat. Biotechnol.*, **27**, 465-471.

[S7] Bintu, L., Buchler, N.E., Garcia, H., Gerland, U., Hwa, T., Kondev, J., and Philips, R. (2005) Transcriptional regulation by the numbers: models. *Curr. Opin. Genet. Dev*., **15**, 116-124.

[S8] Joung, J.K., Koepp, D.M., and Hochschild, A. (1994) Synergistic activation of transcription by bacteriophage lambda cI protein and *E. coli* cAMP receptor protein. *Science*, **265**, 1863-1866.

[S9] Basu, S., Mehreja, R., Thiberge, S., Chen, M., and Weiss, R. (2004)

Spatiotemporal control of gene expression with pulse-generating networks. *Proc. Natl. Acad. Sci. USA*, **101**, 6355-6360.

[S10] Sayut, D.J., Niu, Y., and Sun, L. (2009) Construction and enhancement of a minimal genetic and logic gate. *Appl. Environ. Microbiol.*, **75**, 637-642.

[S11] Guido, N.J., Wang, X., Adalsteinsson, D., McMillen, D., Hasty, J., Cantor, C.R., Elston, T.C., and Collins, J.J. (2006) A bottom-up approach to gene regulation. *Nature*, **439**, 856-860.

[S12] Stricker, J., Cookson, S., Bennett, M.R., Mather, W.H., Tsimring, L.S., and Hasty, J. (2008) A fast, robust and tunable synthetic gene oscillator. *Nature*, **456**, 516-519.

[S13] Atkinson, M.R., Savageau, M.A., Myers, J.T., and Ninfa, A.J. (2003) Development of genetic circuitry exhibiting toggle switch or oscillatory behavior in *Escherichia coli*. *Cell*, **113**, 597-607.

[S14] Cox, R.S. III, Surette, M.G., and Elowitz, M.B. (2007) Programming gene expression with combinatorial promoters. *Mol. Syst. Biol.*, **3**, 145.

[S15] Murphy, K.F., Balázsi, G., and Collins, J.J. (2007) Combinatorial promoter design for engineering noisy gene expression. *Proc. Natl. Acad. Sci. USA*, **104**, 12726-12731.

[S16] Krueger, M., Scholz, O., Wisshak, S., and Hillen, W. (2007) Engineered Tet repressors with recognition specificity for the tetO-4C5G operator variant. *Gene,* **404**, 93-100.

[S17] Isalan, M., Klug, A., and Choo, Y. (2001) A rapid, generally applicable method to engineer zinc fingers illustrated by targeting the HIV-1 promoter. *Nat. Biotechnol*., **19**, 656-660.

[S18] Isaacs, F.J., Dwyer, D.J., Ding, C., Pervouchine, D.D., Cantor, C.R., and

Collins, J.J. (2004) Engineered riboregulators enable post-transcriptional control of gene expression. *Nat. Biotechnol.*, **22**, 841-847.

[S19] Beisel, C.L., Bayer, T.S., Hoff, K.G., and Smolke, C.D. (2008) Model-guided design of ligand-regulated RNAi for programmable control of gene expression. *Mol. Syst. Biol*., **4**, 224.

[S20] Anderson, J., Voigt, C., and Arkin, A. (2007) Environmental signal integration by a modular AND gate. *Mol. Syst. Biol*., **3**, 133.

[S21] Salis, H.M., Mirsky, E.A., and Voigt, C.A. (2009) Automated design of synthetic ribosome binding sites to control protein expression. *Nat. Biotechnol*., **27**, 946-950.

[S22] Kaern, M., Elston, T., Blake, W., and Collins, J.J. (2005) Stochasticity in gene expression: from theories to phenotypes. *Nat. Rev. Genet.*, **6**, 451-464.

[S23] Shea, M.A., and Ackers, G.K. (1985) The OR control system of bacteriophage lambda. A physical-chemical model for gene regulation. *J. Mol. Biol.*, **181**, 211-230.

[S24] Ozbudak, E.M., Thattai, M., Kurtser, I., Grossman, A.D., and van Oudenaarden, A. (2002) Regulation of noise in the expression of a single gene. *Nat. Genetics*, **31**, 69-73.

[S25] Dunlop, M.J., Cox, R.S. III, Levine, J.H., Murray, R.M., and Elowitz, M.B. (2008) Regulatory activity revealed by dynamic correlations in gene expression noise. *Nat. Genetics*, **40**, 1493-1498.

[S26] Elowitz, M.B., Levine, A.J., Siggia, E.D., and Swain, P.S. (2002) Stochastic gene expression in a single cell. *Science*, **297**, 1183-1186.

[S27] Fox, R.F., Gatland, I.R., Roy, R., and Vemuri, G. (1988) Fast, accurate algorithm for numerical simulation of exponentially correlated colored noise. *Pys. Rev. A*, **38**, 5938.