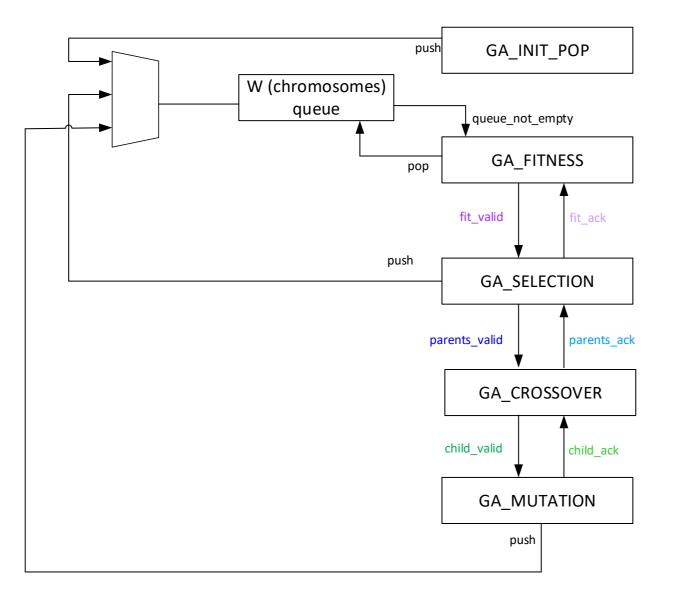


		•	•	•	•	•

	GA_ INIT_POP [Time per chromosome: M/7]	W queue	GA_ FITNESS [Time per chromosome: 13*B]	GA_SELECTION – W sorted pool	GA_SELECTION – parents selection [Time per chromosome: 4]	GA_ CROSSOVER [Time per chromosome: 1]	GA_ MUTATION [Time per chromosome: 1]	
	GA_ INIT_POP	W queue	GA_ FITNESS	GA_SELECTION – W sorted pool	GA_SELECTION – parents selection	GA_ CROSSOVER	GA_ MUTATION	(1)
	GA_ INIT_POP	W queue	GA_ FITNESS	GA_SELECTION – W sorted pool	GA_SELECTION – parents selection	GA_ CROSSOVER	GA_ MUTATION	(2)
	GA_ INIT_POP	W queue	GA_ FITNESS	GA_SELECTION – W sorted pool	GA_SELECTION – parents selection	GA_ CROSSOVER	GA_ MUTATION	(3)
	GA_ INIT_POP	W queue	GA_ FITNESS	GA_SELECTION – W sorted pool	GA_SELECTION – parents selection	GA_ CROSSOVER	GA_ MUTATION	(4)
	GA_ INIT_POP	W queue	GA_ FITNESS	GA_SELECTION – W sorted pool	GA_SELECTION – parents selection	GA_ CROSSOVER	GA_ MUTATION	(5)
• •	GA_ INIT_POP 	W queue	GA_ FITNESS	GA_SELECTION – W sorted pool	GA_SELECTION – parents selection	GA_ CROSSOVER	GA_ MUTATION	(6)
	GA_ INIT_POP •••	W queue	GA_ FITNESS	GA_SELECTION – W sorted pool	GA_SELECTION – parents selection	GA_ CROSSOVER	GA_ MUTATION	(7)
	GA_ INIT_POP	W queue	GA_ FITNESS	GA_SELECTION – W sorted pool	GA_SELECTION – parents selection	GA_ CROSSOVER	GA_ MUTATION	(8)
	GA_ INIT_POP	W queue	GA_ FITNESS	GA_SELECTION – W sorted pool	GA_SELECTION – parents selection	GA_ CROSSOVER	GA_ MUTATION	(9)
	GA_ INIT_POP	W queue	GA_ FITNESS	GA_SELECTION – W sorted pool	GA_SELECTION – parents selection	GA_ CROSSOVER	GA_ MUTATION	(10)
	GA_ INIT_POP	W queue	GA_ FITNESS	GA_SGE!CTION!— W sorteat pool	GA_SELECTION – parents selection	GA_ CROSSOVER	GA_ MUTATION	(11)
	GA_ INIT_POP	W queue	GA_ FITNESS	GA_SELECTION – W sorted pool	GA_SELECTION – parents selection	GA_ CROSSOVER	GA_ MUTATION	(12)
	GA_ INIT_POP	W queue	GA_ FITNESS	GA_SELECTION – W sorted pool	GA_SELECTION – parents selection	GA_ CROSSOVER	GA_ MUTATION	(13)
	GA_ INIT_POP	W queue	GA_ FITNESS	GA_SELECTION – W sorted pool	GA_SELECTION – parents selection	GA_ CROSSOVER	GA_ MUTATION	(14)
	GA_ INIT_POP	W queue	GA_ FITNESS	GA_SELECTION – W sorted pool	GA_SELECTION – parents selection	GA_ CROSSOVER	GA_ MUTATION •	(15)
	GA_ INIT_POP	W queue	GA_ FITNESS	GA_SELECTION – W sorted pool	GA_SELECTION – parents selection	GA_ CROSSOVER	GA_ MUTATION	(16)
	GA_ INIT_POP	W queue	GA_ FITNESS	GA_SELECTION – W sorted pool	GA_SELECTION – parents selection	GA_ CROSSOVER O	GA_ MUTATION	(17)
	GA_ INIT_POP	W queue	GA_ FITNESS	GA_SELECTION – W sorted pool	GA_SELECTION — parents selection	GA_ CROSSOVER <b>○</b>	GA_ MUTATION	(18)
	GA_ INIT_POP	W queue	GA_ FITNESS	GA_SELECTION – W sorted pool	GA_SELECTION – parents selection	GA_ CROSSOVER	GA_ MUTATION	(19)
	GA_ INIT_POP	W queue	GA_ FITNESS	GA_SELECTION – W sorted pool	GA_SELECTION – parents selection	GA_ CROSSOVER O	GA_ MUTATION	(20)
	GA_ INIT_POP	W queue	GA_ FITNESS	GA_SELECTION – W sorted pool	GA_SELECTION – parents selection	GA_ CROSSOVER	GA_ MUTATION	(21)
	GA_ INIT_POP	W queue	GA_ FITNESS	GA_SELECTION – W sorted pool	GA_SELECTION – parents selection	GA_ CROSSOVER	GA_ MUTATION	(22)
	GA_ INIT_POP ···	W queue	GA_ FITNESS	GA_SELECTION – W sorted pool	GA_SELECTION – parents selection	GA_ CROSSOVER	GA_ MUTATION	(23)
	GA_ INIT_POP •••	W queue	GA_ FITNESS	Created pool	GA_SELECTION – parents selection	GA_ CROSSOVER	GA_ MUTATION	(24)
	GA_ INIT_POP	W queue	GA_ FITNESS	GA_SELECTION – W sorted pool	GA_SELECTION – parents selection	GA_ CROSSOVER	GA_ MUTATION	(25)



## MAIN FUNCTION

Gets 42 bit random number at every cylce.
Outputs all of the chromosomes one-by-one
every time a chromosome is ready. Finish when
reaches P chromosomes.

Gets a single individual and outputs it with it's fitness function

Gets inidividuals cycle by cycle and start to sort them. After getting all the individuals, outputs cycle-by-cycle pairs of paraents

Gets a pair of parents and creates a child

Gets a child and makes mutation and output the new individual

## MAIN INTERFACE

[relevant regs: cnfg\_P, cnfg\_M]
Inputs: start\_pls, rand\_42bit
Outputs: init\_pop\_chromosome, push

[relevant regs: cnfg\_B]

[relevant parameters: cnfg\_P]

Inputs: selection\_enable, rand\_42bit[18:0], fit\_chromosome,
fit\_score, fit\_valid, parents\_ack

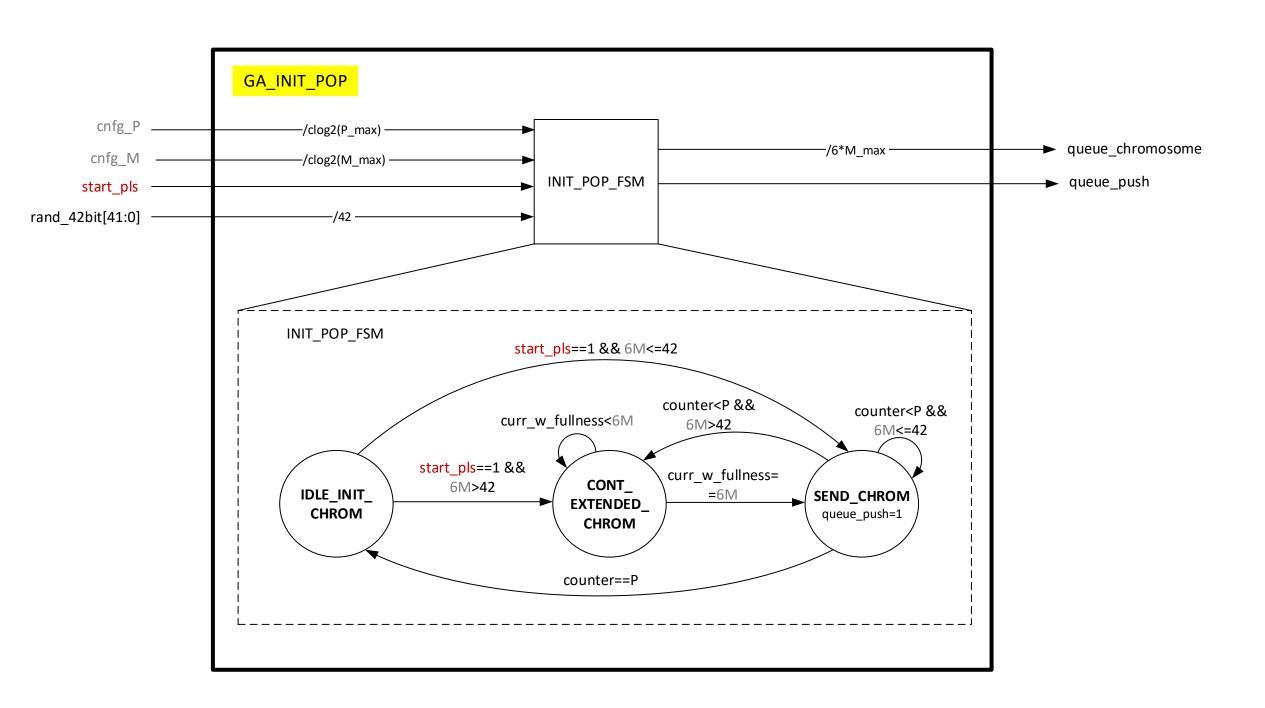
Outputs: gen\_created\_pls, fit\_ack, parent1, parent2, parents\_valid, gen\_best\_chrom, gen\_best\_score, selection\_chromosome, push

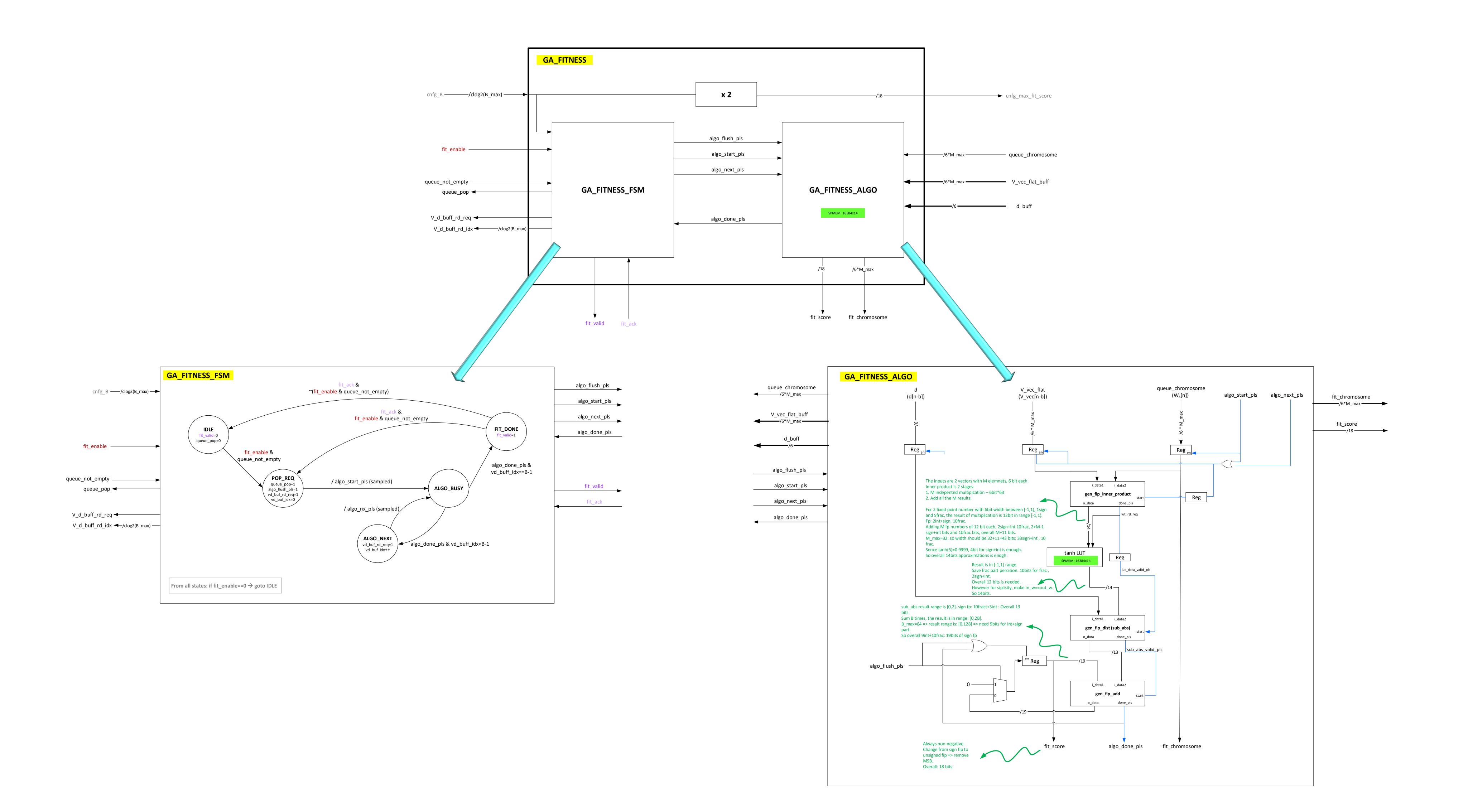
[relevant parameters: cnfg\_M]

Outputs: parents ack, child+child valid

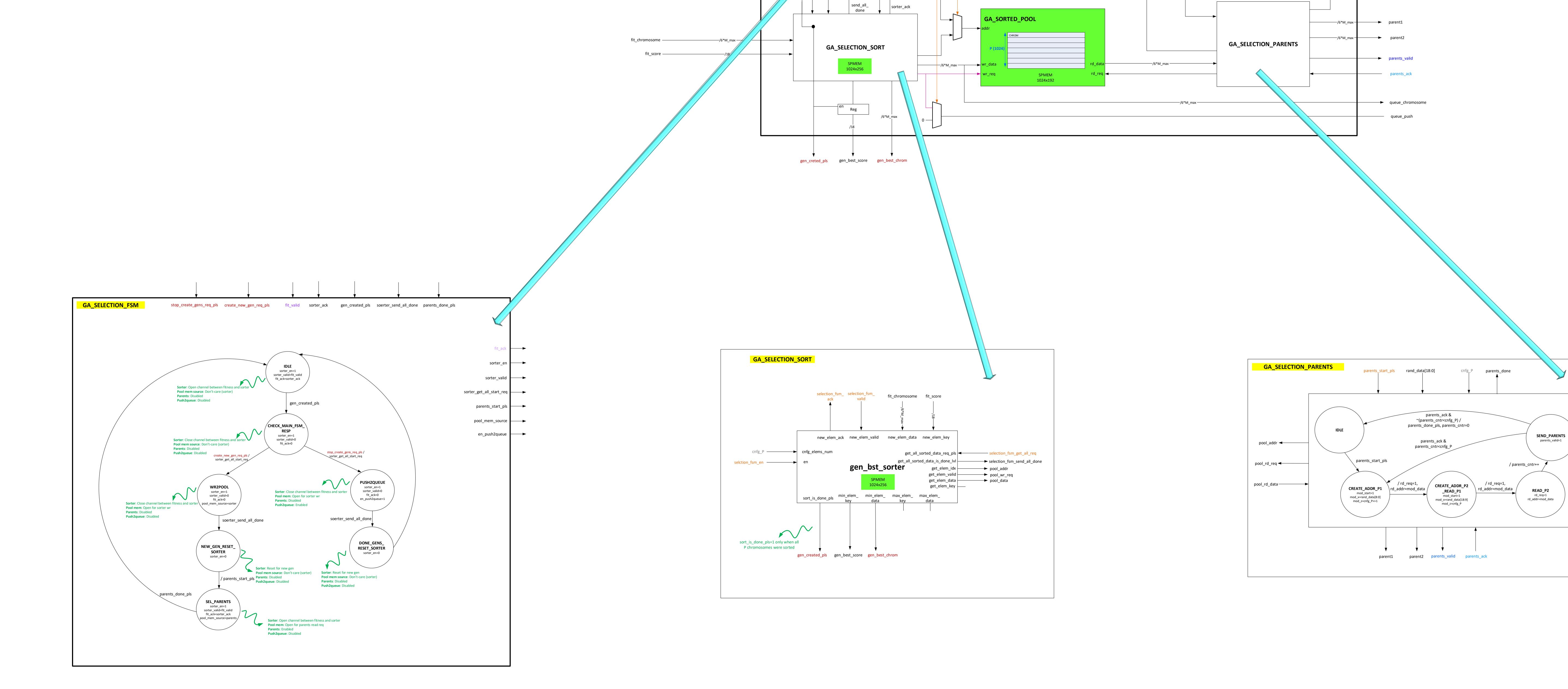
[relevant parameters: cnfg\_M]

Inputs: cnfg\_max\_fit\_score, gen\_best\_score,
 rand\_42bit[39:29], child+child\_valid
 Outputs: child\_ack, chromosome, push





	FETCH {V[b],D[b]} FROM VD BUFF (1clk)	SAMPLE (1clk)	INNER PRODUCT: V dot chromosome (8clk)	Tanh LUT (1clk)	DISTANCE TO d(b) (1clk)	ADD TO ALL PREVIOUS VD (1clk)
(1)	FETCH VD	Sample	INNER PRODUCT	LUT	DIST	ADD
(2)	FETCH VD	Sample	INNER PRODUCT	LUT	DIST	ADD
(3)	FETCH VD	Sample	INNER PRODUCT	LUT	DIST	ADD
(4)	FETCH VD	Sample	INNER PRODUCT	LUT	DIST	ADD
(5)	FETCH VD	Sample	INNER PRODUCT	LUT	DIST	ADD
(6)	FETCH VD	Sample	INNER PRODUCT	LUT	DIST	ADD
(7)	FETCH VD	Sample	INNER PRODUCT	LUT	DIST	ADD
(8)	FETCH VD	Sample	INNER PRODUCT	LUT	DIST	ADD
(9)	FETCH VD	Sample	INNER PRODUCT	LUT	DIST	ADD
(10)	FETCH VD	Sample	INNER PRODUCT	LUT	DIST	ADD
(11)	FETCH VD	Sample	INNER PRODUCT	LUT	DIST	ADD
(12)	FETCH VD	Sample	INNER PRODUCT	LUT	DIST	ADD
(13)	FETCH VD	Sample	INNER PRODUCT	LUT	DIST	ADD •



GA\_SELECTION\_FSM

parents\_done\_pls

cnfg\_P ———————/clog2(P\_max) ————

create\_new\_gen\_req\_pls —————

gen\_created\_pls ◀

fit\_valid ----

stop\_create\_gens\_req\_pls————

